

Johanna Peltoniemi

SOVELLUSVIRTUALISOINNIN MAHDOLLISUUDET JA
HYÖDYNTÄMINEN SATAKUNNAN
AMMATTIKORKEAKOULUSSA
case: Liiketoiminta ja kulttuuri Pori

Tietojenkäsittelyn koulutusohjelma
Ohjelmistosuunnittelun suuntautumisvaihtoehto
2011

SOVELLUSVIRTUALISOINNIN MAHDOLLISUUDET JA HYÖDYNTÄMINEN SATAKUNNAN AMMATTIKORKEAKOULUSSA

case: Liiketoiminta ja kulttuuri Pori

Peltoniemi, Johanna
Satakunnan ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Toukokuu 2011
Ohjaaja: Nuutinen, Petri
Sivumäärä: 53
Liitteitä: 0

Asiasanat: virtualisointi, sovellusvirtualisointi, VMware ThinApp, Microsoft App-V, ohjelmistojen hallinta

Ohjelmistojen hallinta on jokaisessa suuressa organisaatiossa haaste. Samat ohjelmistot täytyy usein asentaa kaikkiin organisaation tietokoneisiin riippumatta siitä, mitä käyttöjärjestelmiä ja ohjelmistoja tietokoneista löytyy ennestään. Ohjelmistoristiriidat ja epäyhteensopivuudet eri käyttöjärjestelmien kanssa ovat suhteellisen yleisiä, mikä vaikeuttaa ylläpitotyön automatisointia. Työn tarkoitus on selvittää, miten ylläpitotoimia voidaan tehostaa sovellusvirtualisoinnin avulla.

Työn tilaaja on Satakunnan ammattikorkeakoulun Liiketoiminta ja kulttuuri Pori, jonka kannalta sovellusvirtualisoinnin mahdollisuuksia tässä työssä tutkitaan. Työssä on otettu tarkasteltavaksi kaksi sovellusvirtualisointiohjelmistoa, VMware ThinApp ja Microsoft App-V, joiden avulla sovellusvirtualisoinnin käyttöönotto olisi mahdollisimman vaivatonta toteuttaa. Työssä tutustutaan sovellusvirtualisoinnin periaatteisiin sekä tutkitaan tarkasteltavana olevien sovellusvirtualisointiohjelmistojen käyttökelpoisuutta niin teorian kautta kuin käytännön testien perusteella.

Sovellusvirtualisoinnin tutkimus aloitetaan perehtymällä perinteiseen ohjelmistojen hallintaan ja sen tuomiin ongelmiin. Tämän jälkeen tutustutaan sovellusvirtualisoinnin peruseräperäisyyteen ja sen tuomiin hyötyihin, haittoihin ja mahdollisuuksiin. Tarkasteltavista sovellusvirtualisointiohjelmistoista käsitellään niiden ominaisuuksia, tietoturvaa ja suorituskykyä teorian tasolla sekä pohjautuen aikaisempiin tutkimuksiin. Lisäksi niiden käyttökelpoisuutta ja toimivuutta vertaillaan käytännön testien avulla. Lopuksi tutkitaan, millaisia mahdollisuuksia sovellusvirtualisointi avaa Satakunnan ammattikorkeakoululle ja mitä asioita tulisi ottaa huomioon sen käyttöönotossa.

Työn tulokset osoittivat, että sovellusvirtualisointi on varteenotettava vaihtoehto tehostaa Satakunnan ammattikorkeakoulun tietokoneiden ylläpitoa. Sen avulla voidaan välttää paljon ohjelmistoristiriitoja sekä yhteensopivuusongelmia eri käyttöjärjestelmäversioiden välillä ja suoraviivaistaa ohjelmistojen jakelua. Sovellusvirtualisoinnin käyttöönotto ei kuitenkaan ole aina ongelmatonta. Satakunnan ammattikorkeakoulun tapauksessa sen toimivuus riippuu siitä, kuinka hyvin koulun ohjelmatarjonta on virtualisoitavissa. Vertailluista ratkaisuvaihtoehdoista VMware ThinApp on parempi vaihtoehto kuin Microsoft App-V, ollen helpompi ja nopeampi käyttää.

APPLICATION VIRTUALIZATION: OPPORTUNITIES AND UTILIZATION IN SATAKUNTA UNIVERSITY OF APPLIED SCIENCES

Case: Faculty of Business and Culture Pori

Peltoniemi, Johanna

Satakunta University of Applied Sciences

Degree Programme in Business Information Systems

May 2011

Supervisor: Nuutinen, Petri

Number of pages: 53

Appendices: 0

Keywords: virtualization, application virtualization, VMware ThinApp, Microsoft App-V, application management

Application management is a challenge in every large organization. Some applications must be usually installed to virtually all computers regardless of the used operating systems and applications that have already been installed. Software conflicts and incompatibilities with different operating systems are common, which makes it difficult to automate the computer maintenance. The purpose of this thesis was to find out how maintenance can be streamlined by the use of application virtualization.

The client of this thesis is Satakunta University of Applied Sciences, Faculty of Business and Culture Pori. This thesis considers two application virtualization solutions, VMware ThinApp ja Microsoft App-V, which are the most viable choices to deploy application virtualization. This thesis introduces the principles of application virtualization and explores the applicability of different application virtualization programs based on both theory and practice.

The study of application virtualization begins by discussing the traditional application management and its problems. After that, the study continues by describing the basic principles, advantages, disadvantages and opportunities of application virtualization. Features, security and performance of the two chosen application virtualization solutions are evaluated in theory. Functionality and user-friendliness of the aforementioned products are compared in practice based on concrete experiments. The thesis ends by studying what kind of opportunities application virtualization can open to Satakunta University of Applied Sciences, and what things should be considered when deploying it.

The results of this thesis prove that application virtualization is considerable option to enhance the computer maintenance of Satakunta University of Applied Sciences. By using application virtualization, many software conflicts and incompatibilities between different operating systems can be avoided. Also the deployment of applications can be streamlined. However, application virtualization is not always trouble-free. Its usability in Satakunta University of Applied Sciences depends on how many of the available applications can be virtualized. The comparison of these two alternatives shows that VMware ThinApp is a better option than Microsoft App-V because it is easier and faster to use.

SISÄLLYS

1	JOHDANTO.....	5
2	PERINTEINEN OHJELMISTOJEN HALLINTA	7
3	SOVELLUSVIRTUALISOINTI.....	10
3.1	Perustietoa sovellusvirtualisoinnista.....	10
3.2	Sovellusvirtualisoinnin hyödyt ja mahdollisuudet.....	12
3.3	Sovellusvirtualisoinnin rajoitukset ja karikot	14
3.4	Paketointivaatimukset.....	14
3.5	Sovellusvirtualisoinnin yhdistäminen muihin tekniikoihin	16
4	SOVELLUSVIRTUALISOINTIOHJELMISTOT	18
4.1	VMware ThinApp.....	18
4.2	Microsoft App-V.....	19
4.3	Sovellusvirtualisoinnin tietoturva	20
4.4	Sovellusvirtualisoinnin suorituskyky.....	22
5	MICROSOFT APP-V JA VMWARE THINAPP VERTAILU	24
5.1	Paketointi VMware ThinApp -ohjelmistolla	24
5.2	Paketointi Microsoft App-V -ohjelmistolla	31
5.3	Käytännön testit ja vertailu	35
5.4	Sovellusvirtualisointiohjelmistojen käytettävyys	39
6	SOVELLUSVIRTUALISOINNIN MAHDOLLISUUDET SATAKUNNAN AMMATTIKORKEAKOULUSSA.....	43
7	SUUNNITTELUNÄKÖKULMAT SOVELLUSVIRTUALISOINNIN KÄYTTÖÖNOTTOON	47
8	YHTEENVETO	49
	LÄHTEET.....	51

1 JOHDANTO

Sovellusten käyttöönotto ja asennus isoissa organisaatioissa on vaativa haaste. Samoja ohjelmistoja täytyy asentaa useisiin eri tietokoneisiin, mikä on erittäin työlästä. Ohjelmistojen yhteensopivuus eri käyttöjärjestelmien välillä tuo ylläpidollisia ongelmia, koska asennustavat voivat erota toisistaan ja joissain tapauksissa vanha ohjelmisto ei edes toimi uudemmissa käyttöjärjestelmissä. Lisäksi ristiriidat eri ohjelmistojen välillä ovat nykyään hyvin yleisiä. Vastaus ylläpidollisiin ongelmiin löytyy sovellusvirtualisoinnista. Sovellusvirtualisointi mahdollistaa sovellusten erottamisen käyttöjärjestelmästä siten, että ne toimivat riippumatta käyttöjärjestelmäversiosta sekä erillään muista ohjelmista.

Työn tilaaja on Satakunnan ammattikorkeakoulu Liiketoiminta ja kulttuuri Pori. Työn tavoitteena on selvittää, kuinka Satakunnan ammattikorkeakoulu voisi tulevaisuudessa hyödyntää sovellusvirtualisointia, jotta tietokoneiden ylläpito olisi nykyistä helpompaa. Työssä tutustutaan VMware ThinApp- ja Microsoft App-V -ohjelmistoratkaisuihin sovellusvirtualisoinnin toteuttamiseksi. Tarkasteltavana olevat sovellusvirtualisointiohjelmistot on valittu Satakunnan ammattikorkeakoulun jo käytettävissä olevista ohjelmistoista, jotta oppilaitos hyötyisi työn tuloksista tulevaisuudessa.

Molempiin tarkasteltavana oleviin sovellusvirtualisointiohjelmistoihin, VMware ThinApp ja Microsoft App-V, tutustutaan sekä teorian että käytännön tasolla. Molemmat ohjelmistot ovat nimenomaan Windows-ympäristössä toimivia ratkaisuja. Tämä tukee Satakunnan ammattikorkeakoulun ylläpitotavoitteita, koska oppilaitoksen IT-laitteisto koostuu enimmäkseen Windows-tietokoneista. Valituista vaihtoehdoista tutkitaan muun muassa niiden soveltuvuutta kouluympäristöön sekä vertaillaan niiden ominaisuuksia ja käytettävyyttä.

Työ on jaettu kuuteen asiakokonaisuuteen. Luvussa 2 tehdään lyhyt katsaus perinteiseen ohjelmistojen hallintaan. Luvussa keskitytään erityisesti tarkastelemaan perinteisen ohjelmistojen hallinnan mukanaan tuomia ylläpidollisia ongelmia suuressa organisaatiossa.

Luvussa 3 tutustutaan sovellusvirtualisoinnin perusperiaatteisiin ja määritellään keskeisimpiä siihen liittyviä käsitteitä. Lisäksi tutkitaan sovellusvirtualisoinnin tuomia hyötyjä ja haittoja, sekä selvitetään, kuinka sovellusvirtualisointi voidaan yhdistää muihin teknologioihin osana organisaation IT-osaston kokonaisratkaisua.

Luvussa 4 tutustutaan kahteen eri sovellusvirtualisointiohjelmistoon: VMware ThinAppiin ja Microsoft App-V:hen. Kummankin sovellusvirtualisointiratkaisun osalta käsitellään sen toimintaperiaatteita ja tärkeimpiä ominaisuuksia. Luvussa perehdytään myös ohjelmistojen hyötyihin ja haittoihin teorian tasolla. Ohjelmistojen tietoturvaa ja suorituskykyä tarkastellaan aikaisempiin tutkimuksiin perustuen.

Luvussa 5 keskitytään vertailemaan VMware ThinAppin ja Microsoft App-V:n käytettävyyttä ja toimivuutta. Vertailu aloitetaan tutustumalla kummankin ohjelmiston paketointiprosesseihin, jonka jälkeen ohjelmistoja vertaillaan käytännön tasolla ajamalla molemmille ohjelmille samanlaiset käyttötestit, joihin perustuen vertaillaan erityisesti paketoinnin onnistumista, pakettien kokoa sekä helppokäyttöisyyttä.

Luvussa 6 selvitetään, kuinka Satakunnan ammattikorkeakoulu voisi mahdollisesti hyötyä sovellusvirtualisoinnin käytöstä. Mahdollisuuksien kartoittamisessa keskitytään erityisesti niihin asioihin, joista Satakunnan ammattikorkeakoulun ylläpito voisi saada paljon hyötyä.

Luvussa 7 käsitellään sovellusvirtualisoinnin käyttöönottoon liittyviä keskeisiä suunnittelunäkökulmia: mitä organisaation tulisi huomioida ja mihin sen tulisi etukäteen varautua sovellusvirtualisoinnin onnistuneen käyttöönoton turvaamiseksi. Luvussa 8 on tehty työn tuloksista yhteenveto.

2 PERINTEINEN OHJELMISTOJEN HALLINTA

Kuka tahansa tietokoneen käyttäjä joutuu asentamaan tietokoneeseen ohjelmistoja. Perinteisesti ohjelmistot ja niiden vaatimat muut ohjelmakomponentit asennetaan käsin, jolloin ohjelmiston toimivuudesta kyseisen tietokoneen käyttöjärjestelmässä varmistutaan välittömästi asennuksen jälkeen. Käsin asentaminen on luotettavin vaihtoehto, ja se on hallittavissa niin yksittäisten ihmisten kuin myös pienien organisaatioiden tapauksessa.

Keskisuurissa tai suurissa organisaatioissa, jotka sisältävät kymmeniä tai jopa satoja tietokoneita, ohjelmistojen asentaminen käsin ei kuitenkaan ole enää järkevä vaihtoehto. Käsin asentamisen sijaan ylläpitotyötä täytyy pystyä automatisoimaan. Perinteisenä ratkaisuna tähän käytetään yleisesti Electronic Software Distribution (ESD) -järjestelmiä. ESD-järjestelmien avulla ohjelmistojen jakelu suoritetaan ilman fyysistä mediaa jostain keskitetystä ohjelmistojen hallinnointipaikasta. Tämä tarkoittaa käytännössä ohjelmistojen jakelua ja asennusta verkon yli. Organisaation ylläpito on vastuussa ohjelmistojen tai päivitysten riippuvuuksien määrittämisestä, asennuspakettien laatimisesta ja niiden laittamisesta jakoon verkossa oleville tietokoneille. (Muller 1998, 315–316.)

ESD-järjestelmä antaa ylläpitäjälle mahdollisuuden määrittää, mihin työpöytätietokoneisiin tai ryhmiin jokin tietty ohjelmisto tai sen päivitys tulee asentaa ja minä päivänä asennus on tarkoitus tehdä. Tämän jälkeen ESD-järjestelmä huolehtii asennuksen toimeenpanosta automaattisesti. (Muller 1998, 315–316.) Esimerkkejä yleisesti käytössä olevista ESD-järjestelmistä Windows-ympäristössä ovat Windows Server Update Services ja Microsoft System Center Configuration Manager. Windows Server Update Services (Elhajj 2005) mahdollistaa päivitysten automatisoidun jakelun. Microsoft System Center Configuration Manager sen sijaan soveltuu yleisesti ohjelmistojen jakeluun.

Kun ohjelmistoja asennetaan perinteisesti ESD-järjestelmien avulla, voi asennuksissa ilmetä tietokonekohtaisesti ongelmia, jotka johtuvat epäyhteensopivuuksista eri käyttöjärjestelmäversioiden kanssa tai ristiriidoista muiden jo tietokoneissa asennettuna

olevien ohjelmistojen kanssa. Ristiriidat muiden ohjelmistojen kanssa johtuvat usein DLL-konflikteista, joita Windows-ympäristössä voi tapahtua hyvinkin helposti, kun eri ohjelmat vaativat samasta DLL-tiedostosta täysin eri versioita. (Ruest & Ruest 2009, 266–277.)

Ainoa varma tapa välttää asennuksen ongelmatilanteet perinteisessä ohjelmistojen hallinnassa on testata jokaisen asennettavan sovelluksen toimivuus etukäteen kaikissa mahdollisissa käyttöjärjestelmä- ja ohjelmistokonfiguraatioissa. Tämä toimenpide täytyy toistaa joka kerta, kun organisaatiossa päivitetään uudempaan käyttöjärjestelmäversioon. Näin ohjelmistojen hallinta voi muodostua hyvin työlääksi, tehottomaksi ja virhealttiiksi. (Ruest & Ruest 2009, 266.)

Ohjelmistoristiriitoja pyritään välttämään yleensä käyttämällä kerroksittaista lähestymistapaa. Ensin kaikkiin tietokoneisiin asennetaan käyttöjärjestelmä ja keskeisimmät työkaluohjelmat, kuten virustorjunta- ja palomuuriohjelmistot. Tämän jälkeen asennetaan ensimmäinen ohjelmistokerros, joka sisältää kaikki ohjelmistot, jotka tulee asentaa organisaation kaikkiin tietokoneisiin. Seuraavaksi asennetaan toinen kerros, joka koostuu ohjelmistoista, jotka tarvitsee asentaa vain osaan organisaation tietokoneista. Tällaisia ohjelmistoja voivat olla esimerkiksi graafisen suunnittelun ohjelmistot, jotka asennetaan vain graafisille suunnittelijoille, ja ohjelmointiympäristöt, jotka asennetaan vain sovelluskehittäjille. Viimeinen ohjelmistokerros jaellaan ad hoc -periaatteella niille, jotka tarvitsevat hyvin tapauskohtaisia sovelluksia. (Ruest & Ruest 2009, 266.)

Eräs tapa vähentää ohjelmistoristiriitoja entisestään on käyttää Windows Installer -palvelua ohjelmistojen asennuksessa. Sovellus paketoidaan MSI-pakettiin, jonka avulla se asennetaan lopulta tietokoneisiin. Windows Installer on suunniteltu ratkaisemaan DLL-konflikteja, mutta muiden konfliktien olemassaolo voi silti estää sovelluksen onnistuneen asennuksen. Windows Installer ei siis kykene estämään muita konflikteja, joita voi syntyä muun muassa tiedostojen, rekisterimerkintöjen, pikakuvakkeiden ja ini-tiedostojen välille. (Ruest & Ruest 2009, 268.)

Perinteisen ohjelmistojen hallinnan ongelmia voidaan havainnollistaa kuvitteellisen esimerkin kautta. Organisaatiossa on sata tietokonetta, joista vanhimmat 60 tietoko-

netta on varustettu Windows XP -käyttöjärjestelmällä ja uusimmat 40 tietokonetta Windows 7 -käyttöjärjestelmällä. Organisaatiossa halutaan ottaa käyttöön eräs grafiikkaohjelmisto, joka on tarkoitus asentaa kaikkiin organisaation tietokoneisiin. Asennettaessa ongelmaksi muodostuu ensinnäkin se, että kyseinen grafiikkaohjelmisto on jo sen verran vanha, että se ei toimi suoraan Windows 7 -käyttöjärjestelmässä vaan vaatii asetuksiin erityisiä muutoksia. Lisäksi puolet Windows XP -tietokoneista sisältävät toisen ohjelmiston, joka on epäyhteensopiva asennettavan grafiikkaohjelmiston kanssa. Näin ollen grafiikkaohjelmiston asennus onnistuu automatisoidusti vain 30 tietokoneeseen. Loppuihin 70 tietokoneeseen grafiikkaohjelmisto on asennettava käsin ja sen asetuksia on säädettävä tietokonekohtaisesti ohjelmiston toimivuuden takaamiseksi.

Esimerkin kaltaisessa tilanteessa organisaatio hyötyy vaivoin automatisoidusta ohjelmistojen jakelusta, koska suurimpaan osaan tietokoneista asennus on suoritettava käsin. Saman ohjelman asentaminen käsin kymmeneen tietokoneeseen on ylläpidon kannalta hyvin työlästä ja turhauttavaa, eikä se palvele organisaation ylläpidon tavoitteita.

3 SOVELLUSVIRTUALISOINTI

Virtualisointi on saavuttanut yhä suurempaa suosiota viime vuosina, koska se tarjoaa monia etuja ohjelmistojen hallintaan erityisesti suuremmille organisaatioille. Moni yritys onkin alkanut kiinnostua virtualisoinnin mahdollisuuksista huomattuaan sen tuoman tehokkaan hyödyn. Virtualisointi on jo suhteellisen vanha idea, mutta se on ehtinyt kehittyä vuosien varrella paljon. Nykyään kaikkien saatavilla onkin lukematon määrä erilaisia virtualisointivaihtoehtoja. Virtualisointi muuttaa oleellisesti tapaa hallita käyttöjärjestelmiä ja sovelluksia, tietoverkkoja ja turvallisuutta. Ruestin mukaan virtualisointi ratkaisee useimpia, jos ei ihan kaikkia ongelmia, joita suurilla organisaatioilla ilmenee ohjelmistojen hallinnassa. (Mäntylä 2008; Ruest & Ruest 2009, XV–XIX; XIX; Hoppe & Seeling 2010.)

Virtualisoinnilla tarkoitetaan tekniikkaa, jonka avulla järjestelmän fyysiset piirteet piilotetaan muilta järjestelmiltä, ohjelmistoilta ja loppukäyttäjiltä. Virtualisointi auttaa hallitsemaan järjestelmiä ja resursseja toiminnallisesti riippumatta niiden ulkoasusta tai tekniikasta. Virtualisointiin kuuluu useita eri osa-alueita. Sovellusvirtualisoinnin lisäksi näitä ovat esimerkiksi työpöytävirtualisointi, palvelinvirtualisointi ja tallennusvirtualisointi. (Mann 2006; Mäntylä 2008.)

Virtualisoinnin merkitys IT-maailmassa kasvaa koko ajan, ja viime vuosien aikana siitä on tullutkin huomattava IT-trendi. Virtualisoinnilla on mahdollista saavuttaa monia etuja, mutta se tuo myös uusia haasteita varsinkin järjestelmien hallintaan. Osa haastetta on se, että on monia eri virtualisointitapoja, joten näiden määritelmät menevät usein sekaisin. (Mann 2006; Mäntylä 2008.)

3.1 Perustietoa sovellusvirtualisoinnista

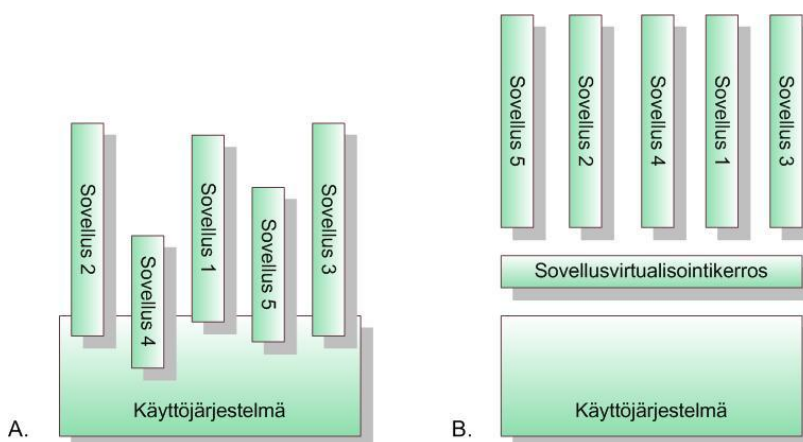
Erityisesti sovellusten hallinnointiin liittyvä virtualisoinnin osa-alue on sovellusvirtualisointi. Sovellusvirtualisointi tarjoaa mahdollisuuden kapseloida ohjelmistot paketteihin, joissa ne suoritetaan itsenäisesti eristettyinä tietokoneen käyttöjärjestelmästä. Sovelluksen paketointi sisältää kaikki asetus-, data- ja ohjelmatiedostot sovelluk-

sen suorittamiseksi, mikä mahdollistaa sovelluksen ajamisen ilman riippuvuuksia. (VMware 2010a.)

Ohjelmistot ovat usein riippuvaisia muista ohjelmista, mikä tarkoittaa sitä, että tietokoneisiin voi joutua asentamaan useita ohjelmia yhden tietyn ohjelmiston toimivuuden takaamiseksi. Suurissa organisaatioissa tämä on työlästä, koska samat asennus- toimenpiteet täytyy suorittaa jokaiseen tietokoneeseen erikseen.

Sovellusvirtualisoinnin avulla on mahdollista asentaa tietokoneelle samojen ohjelmien eri versioita samanaikaisesti. Näin esimerkiksi Access- tai Word-ohjelmistot voidaan asentaa sovellusvirtualisoinnin avulla samaan tietokoneeseen ilman konflikteja. Konfliktit ohjelmien välillä eivät ole mahdollisia, koska ohjelmat on kapseloitu toimimaan erillään käyttöjärjestelmästä ja muista ohjelmista. Jopa ohjelmien käyttäminen saman aikaan ja sisällön kopioiminen ohjelmista toisiin on mahdollista. (Ruest & Ruest 2009, 273.)

Sovellusvirtualisointi muodostaa suojaavan tason käyttöjärjestelmän ja sovelluksen välille. Suojaavan tason sisällä sovellus toimii itsenäisesti riippumatta käyttöjärjestelmästä. Kommunikointi käyttöjärjestelmän kanssa ohjataan suojaavan tason läpi siten, että vain tarpeellinen tieto siirtyy sovelluksen ja käyttöjärjestelmän välillä. (Ruest & Ruest 2009, 273.) Kuvassa 1A on havainnollistettu sovellusten käyttäytyminen ilman sovellusvirtualisointia ja kuvassa 1B sovellusvirtualisoinnin suojaavan tason kanssa.



Kuva 1. Sovellusten käyttäytyminen a) ilman sovellusvirtualisointia b) sovellusvirtualisoinnin kanssa. (Ruest & Ruest 2009, 43.)

Sovellusvirtualisointi voidaan toteuttaa kahdella eri tavalla: agenttipohjaisesti ja agenttittomaksi. Agenttipohjaisessa sovellusvirtualisoinnissa tietokoneeseen täytyy asentaa erillinen agenttiohjelma, joka kykenee ajamaan sovellusvirtualisoituja ohjelmistoja. Agenttittomassa ratkaisussa sen sijaan agenttia ei tarvita, koska agenttiohjelma on sisäänrakennettu itse virtualisoidun ohjelman sisään, jolloin se kulkee aina ohjelman mukana. (Ruest & Ruest 2009, 277.)

Edellä esitetty esimerkki sadan tietokoneen ylläpidosta yksinkertaistuu huomattavasti sovellusvirtualisoinnin ansiosta. Kyseinen grafiikkaohjelmisto paketoidaan sovellusvirtualisoinnin avulla virtualisoituun pakettiin ja sen toimivuus varmennetaan Windows XP -käyttöjärjestelmässä jo paketoituvaiheessa. Tämän jälkeen sovelluksen paketoitu versio voidaan jakaa kaikkiin organisaation sataan tietokoneeseen. Se toimii myös Windows 7 -tietokoneissa käyttöjärjestelmäriippumattomuuden ansiosta. Lisäksi sovellusvirtualisoinnin suojaava kerros estää ohjelmistoristiriitojen syntymisen, joten ohjelmiston asennus onnistuu automatisoidusti kaikkiin organisaation tietokoneisiin. Tilanne on siis täysin toinen kuin käytettäessä perinteisiä ohjelmistojen hallintamenetelmiä.

3.2 Sovellusvirtualisoinnin hyödyt ja mahdollisuudet

Sovellusvirtualisointi tarjoaa monenlaisia hyötyjä ja mahdollisuuksia, joita nykypäivän IT-organisaatiot tavoittelevat. Lisäksi sovellusvirtualisointi mahdollistaa joustavuuden, jota perinteiset hallintatyökalut eivät tarjoa. Sovellusvirtualisoinnilla on useita konkreettisia hyötyjä huolimatta organisaation koosta. Näitä ovat muun muassa hallittavuus, ylläpito ja päivitys keskitetysti, kustannus- ja lisenssisäästöt, tietoturva ja ongelmatilanteista toipuminen. (Mann 2006; Ruest & Ruest 2009, 283–284.)

Hallittavuudella tarkoitetaan sitä, että ohjelmistojen asennus ja ylläpito onnistuvat tavallista helpommin. Sovellusvirtualisoinnissa ohjelma täytyy asentaa vain kerran, jonka jälkeen sen asennus muihin tietokoneisiin vastaa monimutkaisuudeltaan tiedoston kopiointia. Tämä helpottaa asennus- ja ylläpitotoimenpiteitä lyhentämällä asennusaikaa ja vähentämällä virhealttiutta. Varsinkin suurilla ohjelmistoilla on paljon riippuvuuksia toisista ohjelmistoista, jolloin perinteisillä asennusmenetelmillä vas-

taavat asennustoimenpiteet täytyisi suorittaa joka kerta uudelleen. Myös uudelleen-asennus onnistuu tavanomaista paremmin, koska se voidaan suorittaa pelkästään tiedoston kopioinnin avulla. Ohjelmistojen poisto on myös helpommin hallittavissa, koska se tapahtuu yksinkertaisesti poistamalla virtualisoidun ohjelmiston pakettitiedosto tietokoneelta. (Ruest & Ruest 2009, 275–284.)

Sovellusvirtualisointi tuottaa organisaatiossa säästöjä sekä kustannuksissa että lisenssihankinnoissa. Kustannussäästöt syntyvät, kun asennukseen ja ylläpitoon ei tarvitse kuluttaa niin paljon resursseja kuin ilman sovellusvirtualisointia. Ylläpidon työvoimaa voidaan vähentää, mikä tuo kustannussäästöjä jo lyhyellä aikavälillä.

Ohjelmistolisenssien hallitseminen dynaamisesti mahdollistaa mittavat säästöt lisenssihankinnoissa, koska vanhoja lisenssejä voidaan kierrättää tietokoneissa tarpeiden mukaan. Käyttäjän aloittaessa ohjelmiston käytön lisenssipalvelin automaattisesti antaa lisenssipalvelimelta voimassa olevan lisenssin, ja käytön lopetettua lisenssi palautetaan lisenssipalvelimelle. Tällöin ylläpidon ei tarvitse puuttua siihen, mikä lisenssi on jo käytetty ja mitkä lisensseistä ovat yhä käytettävissä. Lisäksi käyttämättömiä lisenssejä on mahdollista kerätä pois ja jakaa myöhemmin uudelleen tarpeen vaatiessa. (Ruest & Ruest 2009, 283–284; SafeNet 2010.)

Sovellusvirtualisointi parantaa tietoturvaa, koska suojaava taso pitää ohjelmiston eristettynä käyttöjärjestelmästä. Näin ollen ohjelmisto ei pääse muokkaamaan käyttöjärjestelmään rekisteriä tai muita asetuksia tai muuttamaan muiden ohjelmistojen tietojen sisältöä. Tämä parantaa erityisesti tiedon eheyttä sovellusvirtualisointia hyödyntävässä tietokoneessa. Ruestin mukaan agenttipohjainen sovellusvirtualisointi ehkäisee lisäksi ohjelmistojen laitonta kopiointia, koska ei riitä, että ohjelmistosta otetaan kopio, vaan tarvitaan myös agenttiohjelma sen ajamista vasten. (Ruest & Ruest 2009, 273–277.)

Sovellusvirtualisointi luo suojaavan tason sovelluksen ja käyttöjärjestelmän ympärille. Koska asennettava sovellus on täysin eristetty käyttöjärjestelmästä ja rekisteristä, mitään muutoksia ei pääse tapahtumaan. Perinteisissä asennuksissa järjestelmä saattaa muuttua kokonaan toimimattomaksi ja tunnistamattomaksi. Tämän takia pitkän ajan kuluessa koneet täytyy asentaa uudelleen ikään kuin vanhaan ajankohtaan alku-

asetuksille, jossa järjestelmä oli normaali. Uudelleenasetaminen taas on työlästä, vie aikaa ja aiheuttaa paljon lisäkustannuksia. (Ruest & Ruest 2009, 42, 273.)

Suojaava taso mahdollistaa myös sen, että virtualisoidut sovellukset voidaan suorittaa missä tahansa käyttöjärjestelmässä ja välttää näin monenlaisia konflikteja myös huonosti toimivien sovellusten välillä. Suojaava taso hallitsee kaikkia tapahtumia virtualisoidun sovelluksen ja Windowsin välillä, ja näin ollen se kykenee muuttamaan epäyhteensopivan sovelluksen yhteensopivaksi eri Windows-versioiden kanssa. (Ruest & Ruest 2009, 273.)

Sovellusvirtualisointi tarvitsee tehdä vain kerran yhdelle sovellukselle. Tämän jälkeen sovellus voidaan kopioida suoraan mihin tahansa tietokoneeseen riippumatta käyttöjärjestelmästä. Käyttöjärjestelmäriippumattomuudesta johtuen sovellusta ei tarvitse paketoida enää uudelleen käyttöjärjestelmän vaihtuessa. (Ruest & Ruest 2009, 42.)

3.3 Sovellusvirtualisoinnin rajoitukset ja karikot

Sovellusvirtualisointiin liittyy muutamia ongelmia. Kaikkia Windows-sovelluksia ei ole kehitetty toimimaan virtuaalisessa ympäristössä. Ongelmien ilmaantuessa ei välttämättä tiedetä heti, johtuvatko ongelmat sovelluksesta itsestään vai aiheutuvatko ne virtuaalisesta ympäristöstä. Sovellusvirtualisointi ei myöskään poista välttämättä kaikkia yhteensopivuusongelmia ja sovellusvirtualisoinnin suojaava kerros voi aiheuttaa lisää ongelmia, mikäli paketoitavassa sovelluksessa on virheitä. Sovellusvirtualisointikerros saattaa myös heikentää jonkin verran suorituskykyä, koska se joutuu ohjaamaan kutsuja käyttöjärjestelmän ja sovelluksien välillä, josta taas aiheutuu lisää kuormitusta. (Jackson; 4Sysops 2008; CSA 2009.)

3.4 Paketointivaatimukset

Useita asioita on otettava huomioon, jotta sovellus voidaan paketoita sovellusvirtualisoinnin avulla onnistuneesti. Ensimmäinen tärkeä vaatimus on, että sovellus toimii paketoituympäristössä. Tämän varmistaminen on kuitenkin suhteellisen helppoa,

koska paketointi voidaan suorittaa virtuaalikoneeseen asennetussa käyttöjärjestelmässä (Ruest & Ruest 2009, 294). Virtuaalikoneen käyttöjärjestelmä voidaan valita sovelluskohtaisesti siten, että paketoitava sovellus toimii täysin oikein kyseisessä käyttöjärjestelmässä. Näin varmistutaan siitä, että myös virtualisoidusta paketista tulee toimiva.

VMwaren mukaan sovelluksen paketointi on syytä suorittaa mahdollisimman puhtaassa tietokoneessa, jotta paketointiprosessi nauhoittaa varmasti kaikki sovelluksen vaatimat muutokset järjestelmään. Puhdas tietokone tarkoittaa käytännössä sitä, että tietokoneeseen on asennettu vain Windows-käyttöjärjestelmä eikä mitään ylimääräisiä ohjelmistoja. Mikäli puhdasta tietokonetta ei käytetä paketoinnissa, asennusohjelma yleensä ohittaa tiedostot, jotka tietokoneesta löytyy jo ennestään. Tällöin sovellusvirtualisoitu sovellus ei välttämättä toimi ollenkaan tai on riippuvainen käyttöjärjestelmään jo asennetuista ohjelmistoista, mikä heikentää paketoitun sovellusversion toimintavarmuutta. (VMware 2010b, 12.)

Puhtaan tietokoneen käytöllä on se seuraus, että paketointiprosessissa on asennettava kaikki sovelluksen vaatimat ohjelmakomponentit. Ylläpitäjän täytyy siis selvittää, mistä ohjelmakomponenteista sovellus on riippuvainen. Esimerkiksi useat sovellukset vaativat nykyään ajonaikaisen Visual C++ Redistributable -ohjelmakomponentin asennuksen.

Sovelluksesta on myös selvitettävä, toimiiko se jollain tapaa epästandardinmukaisella tavalla. Epästandardinmukaisesti toimivat sovellukset voivat hyvinkin olla toimimatta paketoituina, koska sovellusvirtualisointi antaa sovellukselle pääsyn käyttöjärjestelmään vain standardin rajapinnan kautta (Ruest & Ruest 2009, 273). Epästandardinmukaisuuden selvittäminen etukäteen on kuitenkin hankalaa, koska sovelluksen sisäinen toteutus ei ole ylläpitäjän tiedossa. Ainoa tapa varmistaa sovelluksen toimivuus on paketoita ja testata se.

3.5 Sovellusvirtualisoinnin yhdistäminen muihin tekniikoihin

Sovellusvirtualisointi voi usein olla vain yksi osa organisaation suurempaa ylläpidollista kokonaisratkaisua. Koska sovellusvirtualisointi ratkaisee vain tietyt ongelmat ylläpidossa, yhdistämällä muita tekniikoita sen kanssa voidaan saavuttaa entistä tehokkaampi ja toimivampi ylläpito etenkin suurissa organisaatioissa.

Sovellusvirtualisointi voidaan helposti yhdistää sovellusten suoratoistoon (engl. streaming). Suoratoiston avulla sovellukset voidaan käynnistää verkon yli, vaikka niitä ei ole asennettu paikallisesti käyttäjälle. Käyttäjillä on kuitenkin aina pääsy sovelluksiin, vaikka heillä ei olisikaan nettiyhteyttä juuri sillä hetkellä, koska niistä on otettu paikallinen kopio tietokoneelle. Sovelluksen päivittämiseksi täytyy vain luoda päivityspaketti, jonka avulla sovellus päivittyy automaattisesti seuraavan kerran, kun käyttäjä käynnistää kyseisen sovelluksen. (Ruest & Ruest 2009, 42; 283–284.)

Sovellusvirtualisoinnin yhdistäminen muihin virtualisointitekniikoihin on otettu niitä suunniteltaessa varsin hyvin huomioon. Yhdistämällä sovellusvirtualisointi profiilivirtualisoinnin ja työpöytävirtualisoinnin kanssa saadaan aikaiseksi täysin uudenlainen tapa työpöytä-tietokoneiden hallintaan (Ruest & Ruest 2009, 296).

Profiilivirtualisoinnin tarkoituksena on huolehtia käyttäjien profiilien hallinnoinnista siten, että kunkin käyttäjän tekemät asetukset eri sovelluksiin säilyvät kaikista järjestelmässä tapahtuvista muutoksista huolimatta. Profiilivirtualisointi pyrkii erityisesti varmistamaan sen, että kunkin käyttäjän omat tiedot on suojattu siten, että ne eivät sotke toisten käyttäjien tietoja tai eivät joudu väärän henkilön nähtäville. Näin taataan tietojen eheys ja luottamuksellisuus. Microsoft Windowsin käyttäjäprofiilien hallinnointi toteuttaa jo itsessään profiilien virtualisointia luomalla tietokoneeseen automaattisesti paikallisen profiilin kullekin käyttäjälle, kun käyttäjä kirjautuu ensimmäistä kertaa kyseiselle tietokoneelle. Tämä onnistuu erityisen hyvin yhdistettynä Active Directory -autentikointiin, jonka avulla samojen käyttäjätunnusten käyttö voidaan sallia missä tahansa saman verkon tietokoneessa. (Ruest & Ruest 2009, 299.) Microsoft Windowsin sisäänrakennetun ominaisuuden lisäksi virtualisointiohjelmistojen valmistajat tarjoavat profiilivirtualisoinnin toteuttamiseen myös edistykselli-

sempiä ratkaisuja, kuten VMwaren kehittämä VMware View -ohjelmisto (VMware 2007).

Työpöytävirtualisointi sen sijaan korvaa käyttäjän tietokoneen virtuaalisella tietokoneella. Käyttäjän virtuaalinen tietokone sijaitsee organisaation keskitetyssä datakeskuksessa, josta sitä voidaan käyttää etäyhteyden avulla miltä tahansa päätetietokoneelta. Päätetietokone voi olla tavallinen työpöytätietokone tai erityisesti etäyhteyden käyttöön tarkoitettu desktop-client-tietokone, joka sisältää vain minimaalisen määrän resursseja eikä erillistä käyttöjärjestelmää. Päätetietokonetta on tarkoitus käyttää vain etäyhteyden muodostamiseen virtuaalikoneeseen sekä syöte- ja näyttölaitteena, joten siihen ei tarvitse olla asennettuna mitään ohjelmistoja pääteohjelman lisäksi.

(Petrović & Fertalj 2009, 241.)

Sovellusvirtualisoinnin, profiilivirtualisoinnin ja työpöytävirtualisoinnin sekä suoratoiston yhdistämisellä saavutetaan useita etuja. Sovellukset voidaan paketoita sovellusvirtualisoinnin avulla ja siirtää suoratoistopalvelimelle, josta ne voidaan keskitetysti jakaa mihin tahansa tietokoneeseen, joka kuuluu organisaation tietoverkkoon. Kunkin käyttäjän tiedot voidaan suojella profiilivirtualisoinnin avulla, jolloin käyttäjän tiedot pysyvät tietokoneella eheinä ja luottamuksellisina laajastakin käyttäjämäärästä huolimatta. Tämä on myös erittäin tärkeää tuottavuuden kannalta, jotta käyttäjien aikaa ei kulu tarpeettomasti asetusten muokkaamiseen. Työpöytävirtualisointi tuo lisäksi käyttäjille mahdollisuuden tehdä töitä ensin muualla, jonka jälkeen sammuttamatta ohjelmia siirtyä toiselle työpisteelle ja jatkaa työskentelyä täsmälleen siitä, mihin jäi ensimmäisellä työpisteellä. (Ruest & Ruest 2009, 296–299.)

Mitkään edellä esitellyistä tekniikoista eivät kuitenkaan ratkaise laitteistoajurien, käyttöjärjestelmän päivitysten sekä matalan tason suojaohjelmistojen, kuten palomuurin ja virustorjunnan, helppoa asentamista. Nämä täytyy yleensä asentaa tietokoneisiin perinteisellä tavalla, koska matalan tason palveluja ei pystytä paketoimaan kovin hyvin sovellusvirtualisoinnin avulla. Kyseisten järjestelmä- ja ohjelmistokomponenttien asennukseen voidaan organisaation kokonaisratkaisussa hyödyntää ESD -järjestelmiä. (Ruest & Ruest 2009, 273–297.)

4 SOVELLUSVIRTUALISOINTIOHJELMISTOT

Sovellusvirtualisointia varten vaaditaan siihen soveltuva ohjelmisto. Hyvä sovellusvirtualisointiohjelmisto tarjoaa helppokäyttöisen tavan paketoita sovelluksia virtualisoiduiksi packageiksi. Ylläpidon kannalta on erityisen tärkeää, että paketointi ei vie myöskään liian paljon aikaa. Muuten sovellusvirtualisoinnista saatavat säästöt eivät olisi merkittäviä.

Käytön kannalta on myös hyvä, että sovelluksen paketoitu koko pysyy mahdollisimman pienenä. Tämä on tärkeää sekä kiintolevytilaa ajatellen että etenkin, jos ohjelmistoa on tarkoitus jakaa verkon yli suoratoistoa käyttäen. Lisäksi täytyy kiinnittää huomiota siihen, että sovellusvirtualisointi toimii yhteensopivasti eri ohjelmistojen kanssa.

Useilta eri valmistajilta löytyy ratkaisuja sovellusvirtualisoinnin toteuttamiseksi. Kaupallisista sovellusvirtualisointiohjelmistoista tunnetuimpia ovat VMware ThinApp ja Microsoft App-V. Muita kaupallisia vaihtoehtoja ovat muun muassa Citrix XenApp, InstallFree Bridge ja Symantec Software Virtualization Solution Pro. (Ruest & Ruest 2009, 42–43.) Kaupallisten ohjelmistojen lisäksi markkinoilta löytyy myös yksi ilmainen vaihtoehto, Cameyo (Cameyon [www-sivut](http://www.cameyo.com) 2011). Tässä työssä keskitytään kuitenkin tarkastelemaan vain VMware ThinApp - ja Microsoft App-V - ohjelmistoja, jotka soveltuvat hyvin sovellusvirtualisointiin Microsoft Windows - ympäristössä ja ovat työn tilaajana olevan Satakunnan ammattikorkeakoulun Liiketoiminta ja kulttuuri Pori kannalta potentiaalisimmat vaihtoehdot.

4.1 VMware ThinApp

VMware ThinApp (ennen Thinstall) on VMwaren tuoteperheeseen kuuluva ohjelmisto sovellusvirtualisoinnin toteuttamiseen. VMware ThinApp käyttää agenttointia sovellusvirtualisointia, joten ThinApp-ohjelmistolla paketoitujen sovellusten jakaminen ei vaadi erillisen agenttiohjelman asentamista käyttäjän koneelle. (Ruest & Ruest 2009, 292.)

ThinApp koostuu käytännössä vain paketoitISOvelluksesta. Paketoitiproessin jälkeen tuloksena ovat sovelluksen datatiedostot ja asetukset sisältävä dat-tiedosto sekä exe-ohjelmatiedosto, josta paketoitu sovellus saadaan paketoinnin jälkeen käynnistettyä suoraan. VMware ThinApp itsessään ei sisällä mitään jakelutekniikkaa, mutta sen luomat sovelluspaketit on valmiiksi optimoitu suoratoistoa varten. Agentittomuudesta on se etu, että paketoititut sovellukset voidaan jakaa työpöytäietokoneille käyttäen perinteisiä ESD-jakelujärjestelmiä. Näin voidaan hyödyntää samaa jakeluperiaatetta sekä paketoituihin että paketoimattomiin sovelluksiin. Agentittomuuden haittapuolella on kuitenkin se, että paketoituja sovelluksia on hyvin helppo varastaa. Tämä täytyy estää toteuttamalla erityisiä suojamenetelmiä, jotta ohjelmistovarkauksilta vältytään. (Ruest & Ruest 2009, 292.)

ThinApp sisältää paketoitujen sovellusten päivitystä varten ohjelmakomponentin nimeltä AppSync. AppSync päivittää automaattisesti paketoititut sovellukset, kun sovelluksen ini-asetustiedostoon on määritetty lähdeosoite, josta päivityksiä tulee haakea. (Ruest & Ruest, 292.)

4.2 Microsoft App-V

Microsoft App-V on kokonainen tuoteperhe sovellusvirtualisoinnin tarpeisiin. Se sisältää sekä paketointiin käytettävän Microsoft Application Virtualization Sequencer -ohjelman, suoratoiston toteuttamiseen vaadittavat palvelinpuolen ohjelmistot että asiakaspuolen Microsoft Application Virtualization Client -ohjelman.

Microsoft App-V tarjoaa sovellusvirtualisoidujen sovellusten käyttöönottoon kolme eri vaihtoehtoa. Sovellusvirtualisoidut sovellukset voidaan jakaa suoratoiston avulla työpisteille hyödyntäen App-V:n kaikkia ominaisuuksia käyttämällä Microsoft App-V:hen kuuluvaa System Center Application Virtualization Management Server -palvelinohjelmistoa. System Center Application Virtualization Management Server hyödyntää Active Directory -järjestelmää käyttäjien hallinnointiin ja käyttää SQL Server -tietokantaa Microsoft Application Virtualization Database -tietokannan tallentamiskeinona. (Ruest & Ruest 2009; 291; Microsoft 2011.)

Toinen vaihtoehto sovellusten käyttöönottoon on käyttää kevyempää System Center Application Virtualization Streaming Server -ohjelmistoa, joka jakaa ohjelmistot yhä suoratoiston avulla mutta ei hyödynnä Active Directory -järjestelmää. Kolmas vaihtoehto on käyttää Microsoft App-V:tä standalone-tilassa, jolloin sovellusvirtualisoiduja sovelluspaketteja voidaan käynnistää paikallisesti yhdeltä tietokoneelta ilman suoratoistoa ja sen vaatimaa monimutkaista palvelinverkostoa. (Ruest & Ruest 2009, 291.)

Toisin kuin VMware ThinApp, Microsoft App-V hyödyntää agenttipohjaista sovellusvirtualisointia. Se siis vaatii, että käyttäjän koneeseen asennetaan Microsoft Application Virtualization Client -agenttiohjelma, joka huolehtii virtualisoitujen sovellusten käynnistämisestä. Agenttiohjelma voidaan asettaa toimimaan suoratoistotilassa tai standalone-tilassa riippuen siitä, mitä tapaa halutaan käyttää sovellusten jakeluun. (Ruest & Ruest 2009, 289–291.)

Agenttipohjaisuudesta johtuen Microsoft App-V kärsii samoista ongelmista kuin mikä tahansa muu agenttipohjainen sovellusvirtualisointiratkaisu. Sen käyttöönotto on työläämpää kuin agentittoman sovellusvirtualisoinnin, koska jokaiseen tietokoneeseen täytyy asentaa agenttiohjelma, jotta paketoituja versioita pystytään ylipäätään käynnistämään. Toisaalta Microsoft App-V:n agenttipohjaisuus suojaa ohjelmia väärinkäytöksiltä. (Ruest & Ruest 2009, 277.)

4.3 Sovellusvirtualisoinnin tietoturva

Sovellusvirtualisoinnissa tietoturva-aukkoja on mahdollista hyödyntää monella eri tavalla. Sovellusvirtualisointi ei poista ohjelmistoissa olevien haavoittuvuuksien uhkaa. Hyökkääjät voivat hyödyntää ohjelmistojen haavoittuvuuksia samalla tapaa kuin ennenkin. Sovellusvirtualisoitujen ohjelmistojen tietoturvassa on eroja riippuen siitä, ajetaanko niitä normaalin käyttäjän oikeuksin vai järjestelmävalvojan oikeuksin. (Hoppe & Seeling 2010, 294–295.)

Normaalissa käyttäjätilissä haavoittuvuuksia hyödyntäessään hyökkääjä saa vastaatvat oikeudet ja pääsyn resursseihin, jotka käyttäjälläkin normaalisti on. Hyökkääjä

pystyy myös tarkkailemaan käyttäjän toimia käyttämällä etäyhteyteen DLL-injektiohyökkäystä, jossa haitallista koodia sisältävä DLL-tiedosto pakotetaan latautumaan prosessin osoiteavaruudessa. Tämän jälkeen DLL-tiedosto toimii kuten mikä tahansa muu dynaamisesti linkitetty kirjasto ja pääsee ajamaan haitallista koodia ilman, että käyttäjä erikseen käynnistäisi viruksen sisältävän ohjelman. Yhteys on kuitenkin mahdollista katkaista sulkemalla kyseinen virtualisoitu sovellus. (Turkulainen 2004; Hoppe & Seeling 2010, 294–295.)

Järjestelmävalvojan oikeuksin käynnistetty virtualisoitu ohjelma saa automaattisesti järjestelmävalvojan oikeudet, jolloin mahdollinen hyökkääjä saattaa käyttää tietoturva-aukkoa ja saa myös järjestelmävalvojan oikeudet. Järjestelmävalvojan oikeuksilla hyökkääjän on mahdollista muuttaa, luoda tai poistaa tietoja kyseisessä työasemassa. Järjestelmävalvojan oikeudet tuo hyökkääjälle mahdollisuuden myös muokata tai vahingoittaa virtualisoitua sovellusta virtualisoidussa käyttöjärjestelmäympäristössä. (Hoppe & Seeling 2010, 294–295.)

Sovellusvirtualisoinnin tietoturvassa on eroja agentittoman ja agenttipohjaisen menetelmän välillä. Agenttipohjaisessa menetelmässä on suojausmekanismi, koska virtualisoidut sovellukset eivät käynnisty ilman saatavilla olevaa agenttiohjelmaa. Jos sovellusvirtualisointiohjelmisto on agenttipohjainen, kaikki sovellukset on suojattu. Agenttipohjaisessa sovellusvirtualisoinnissa agenttiohjelma ja sovellusvirtualisoitu sovellus ovat erikseen, joten pelkällä sovellusvirtualisoidulla sovelluksella ei yksinään tee vielä mitään. Sen sijaan agentittomat sovellukset ovat täysin siirrettävissä eli käyttäjä pystyy kopioimaan agentittoman sovelluksen ja suorittaa sen missä tahansa järjestelmässä. Sovelluksia on pyrittävä suojaamaan muillakin keinoin lisenssien leviämisen estämiseksi. (Ruest & Ruest 2009, 277–278.)

Virtualisoiduissa sovelluksissa ei kannata unohtaa tietoturvan pohtimista. Koska virtualisoitu sovellus on aina jollain tapaa vuorovaikutuksessa isäntäkäyttöjärjestelmän kanssa, on se haavoittuvainen tietoturvaongelmille. Virtualisoiduissa sovelluksissa on siis samat riskit kuin sovelluksissa, jotka on asennettu työasemaan perinteisellä tavalla. Virtualisoitujen sovellusten määrän kasvaessa on erittäin tärkeää käyttää samoja yleisiä tietoturvatapoja ja -menetelmiä, joita ei pysty millään tavalla laiminlyömään tai ohittamaan. (Hoppe & Seeling 2010, 294–295.)

4.4 Sovellusvirtualisoinnin suorituskyky

Sovellusvirtualisoinnin käyttökelpoisuuden kannalta on tärkeää vertailla, kuinka suorituskykyisiä eri sovellusvirtualisointiohjelmistoilla paketoitua sovelluksia ovat. Yleensäkin on tärkeää tutkia, onko paketoitua sovelluksen suorituskyky riittävä verrattuna paketoimattoman suorituskykyyn. Sovellusvirtualisoinnilla paketoitujen sovellusten tulisi olla riittävän suorituskykyisiä taatakseen sovellusten sulavan käytön kaikissa tilanteissa.

Competitive System Analysis -yrityksen tekemässä tutkimuksessa vertailtiin usean eri valmistajan sovellusvirtualisointiratkaisujen tuottamaa suorituskykyä niin suoritusajan, muistinkäytön kuin suoratoiston verkkokäytön osalta. Tutkimuksessa vertailtujen sovellusvirtualisointiohjelmistojen joukkoon kuuluivat myös tässä työssä vertailtavat VMware ThinApp ja Microsoft App-V. Lisäksi vertailukohtana testeissä käytettiin paketoimattomaa sovellusversiota. CSA käytti testiohjelmana DMS Clarity Suite OfficeBench -testiskriptiä, joka käyttää OLE-automaatiota ajaakseen joukon simuloituja testejä Microsoft Office -tuoteperheen sovelluksilla sekä Internet Explorer -selaimella (CSA 2009.) Näin ollen tutkimuksen testitapaus koostui päivittäisessä käytössä olevista sovelluksista, joiden sulava toiminta on erittäin tärkeää käyttäjien tuottavuuden kannalta.

CSA:n tutkimuksen mukaan OfficeBench-sovellus oli suoritusajan osalta VMware ThinApp -ohjelmistolla paketoituna noin 24 % nopeampi kuin Microsoft App-V:llä paketoituna. Syy tälle on se, että Microsoft App-V:n käyttämä erillinen agenttiohjelma aiheuttaa sovellukselle ylimääräistä laskennallista taakkaa verrattuna ThinAppiin, jossa agenttiohjelma on suoraan sovelluspaketin sisällä. Erillinen agenttiohjelma oli myös syy sille, että Microsoft App-V:llä paketoitu versio Microsoft Office -ohjelmistosta kulutti yli kaksinkertaisen määrän muistia verrattuna ThinAppilla paketoitua versioon (CSA 2009.) Näin suuri ero muistinkäytössä voi tulla ratkaisevaksi etenkin siinä vaiheessa, jos halutaan paketoita sovelluksia, joiden muistinkulutus on gigatavujen luokkaa. Microsoft App-V kulutti myös noin 50 % enemmän verkkokaistaa verrattuna ThinAppiin, kun suoratoistoa käytettiin sovelluksen jakeluun käyttäjän tietokoneelle (CSA 2009).

Kumpikaan sovellusvirtualisointiratkaisu ei kuitenkaan yltänyt CSA:n testeissä paketoimattomana ajetun sovelluksen suorituskykyyn. VMware ThinApp, jonka suorituskyky vastasi eniten paketoimattoman version suorituskykyä, oli sekin suoritusajassa mitattuna lähes 30 % paketoimatonta versiota hitaampi. (CSA 2009.) Suurta suorituskykyä vaativien sovellusten osalta voi ollakin, että niiden paketointi sovellusvirtualisoinnin avulla ei ole järkevää. Huonommasta suoritusajasta huolimatta ThinAppilla paketoitun sovelluksen muistinkäyttö oli vain hieman suurempi kuin paketoimattoman version (CSA 2009).

CSA:n testien perusteella voidaan sanoa, että VMware ThinApp on selvästi Microsoft App-V:tä suotuisampi vaihtoehto, mikäli sovelluksen suorituskyvyllä on paljon merkitystä. Selvästi paremman suoritustehon lisäksi ThinAppilla paketoitut sovellukset rasittavat paljon vähemmän verkkoa suoratoistoa käytettäessä ja käyttävät huomattavasti vähemmän muistia kuin mitä ne tekisivät Microsoft App-V:llä paketoituna.

5 MICROSOFT APP-V JA VMWARE THINAPP VERTAILU

Sovellusvirtualisointiohjelmistojen tekniset ominaisuudet itsessään eivät riitä ratkaisuun, kumpi tarkasteltavana olevista ohjelmistoista, VMware ThinApp vai Microsoft App-V, on parempi vaihtoehto organisaation ohjelmistojen ylläpidon ratkaisuksi. Tärkeämpää on tutkia käytännön tasolla ohjelmistojen eroja sekä niiden mukanaan tuomia hyötyjä ja haittoja.

Vertailun kannalta on tärkeää tutkia, millainen paketointiprosessi on kummallakin sovellusvirtualisointiohjelmistolla ja kuinka yhteneviä tai eroavia ne ovat. Paketointiprosessin paremmuutta voidaan arvioida erityisesti helppokäyttöisyyden ja ajankäytön kannalta. Sovellusvirtualisointiohjelmistojen toimivuuden analysoimiseksi ja käytännön vertailua varten suoritettiin myös käytännön testejä kummallakin sovellusvirtualisointiohjelmistolla.

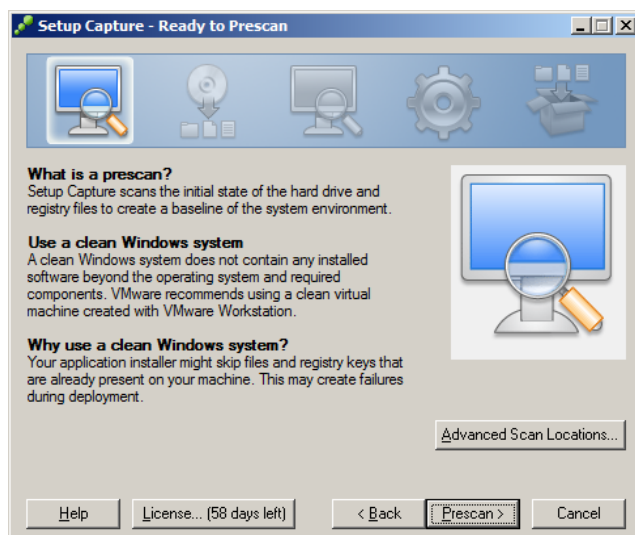
5.1 Paketointi VMware ThinApp -ohjelmistolla

VMware ThinApp -ohjelmiston asentaminen on nopeaa ja vaivatonta, koska asennuspaketti on kooltaan varsin pieni. Tämä tarkoittaa myös sitä, että ThinApp vie varsin vähän tilaa tietokoneelta. Sovelluksen paketointi aloitetaan käynnistämällä ThinApp Setup Capture -ohjelma. Kuvassa 2 on esitetty ThinApp-ohjelmiston aloitusruutu.



Kuva 2. VMware ThinApp -ohjelmiston aloitusruutu.

Ensimmäinen tärkeä vaihe paketoinnissa on esiskannaus (engl. prescan), joka on esitetty kuvassa 3. Tässä vaiheessa ThinApp skannaa Windowsin rekisterin sekä valitut kiintolevyasemat, jotta se tietää, missä tilassa tietokone on rekisterin ja tiedostojen osalta ennen asennusta. Tämä on välttämätöntä, jotta ThinApp kykenee jatkossa selvittämään paketoitavan sovelluksen asentamisessa tapahtuvat muutokset.



Kuva 3. ThinApp-ohjelmiston esiskannausvaihe.

Esiskannauksen jälkeen ThinApp on valmis paketoitavan sovelluksen asentamista varten (kuva 4). Tässä vaiheessa sovellus asennetaan samalla tavalla kuin se tehtäisiin perinteisellä tavallakin. Mikäli sovellus vaatii muita sovelluksia tai kirjastoja toimiakseen, asennetaan myös ne tässä vaiheessa. Lopuksi sovellus käynnistetään, jotta se voi suorittaa mahdolliset alkukonfiguraatiot, jotka voivat vaikuttaa tietokoneen rekisterin tilaan tai tiedostoihin. Halutessaan sovelluksen asetuksia voidaan muuttaa myös tässä vaiheessa, jotta ne siirtyvät suoraan sovellusvirtualisoituun versioon sovelluksesta.



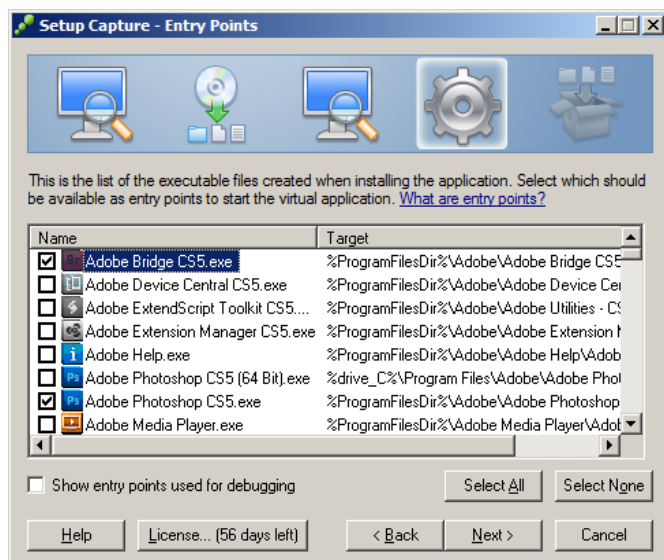
Kuva 4. ThinApp valmiina sovelluksen asentamista varten.

Sovelluksen asentamisen jälkeen voidaan siirtyä jälkiskannausvaiheeseen (engl. postscan), joka on esitetty kuvassa 5. Jälkiskannausvaiheessa ThinApp skannaa Windowsin rekisterin ja kiintolevyasemat samalla tavalla kuin esiskannausvaiheessa. Lopuksi ThinApp vertaa jälkiskannauksen tilaa esiskannauksen tilaan ja päättelee, mitkä tiedostot ja rekisteriarvot lisättiin tai muuttuivat.



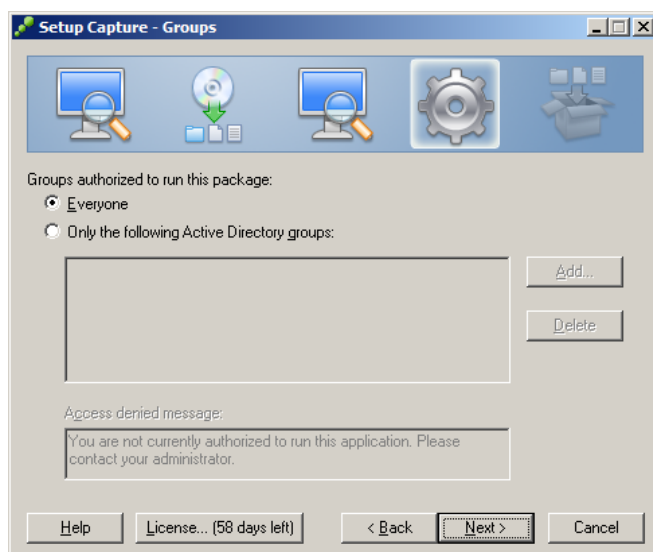
Kuva 5. ThinApp jälkiskannausvaiheessa.

Jälkiskannausvaiheen jälkeen valitaan, mitkä asennetut exe-ohjelmatiedostot on mahdollista käynnistää sovellusvirtualisoidun paketin kautta (kuva 6). Tämä täytyy tehdä, koska ThinApp-ohjelmisto ei pysty itse pääättelemään, mitkä exe-ohjelmatiedostoista ovat oleellisia.



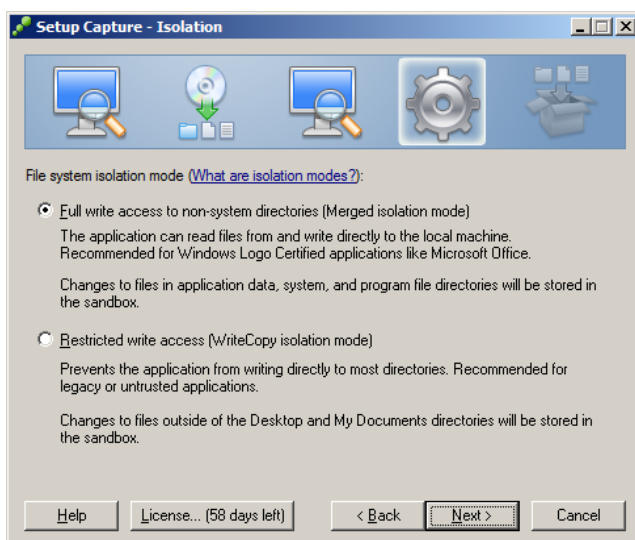
Kuva 6. Ohjelmatiedostojen valitseminen.

Seuraavaksi ThinApp-ohjelmistolle kerrotaan, ketkä käyttäjät saavat ajaa sovellusvirtualisoitua sovellusta (kuva 7). Oikeudet voidaan antaa joko kaikille käyttäjille tai valituille Active Directory -käyttäjärhyhmille. Käyttäjärhyhmien valinnalla voidaan ennaltaehkäistä tietoturvaongelmia ja rajoittaa erikoisemmat sovellukset vain niiden käyttöön, joilla siihen on oikeus.



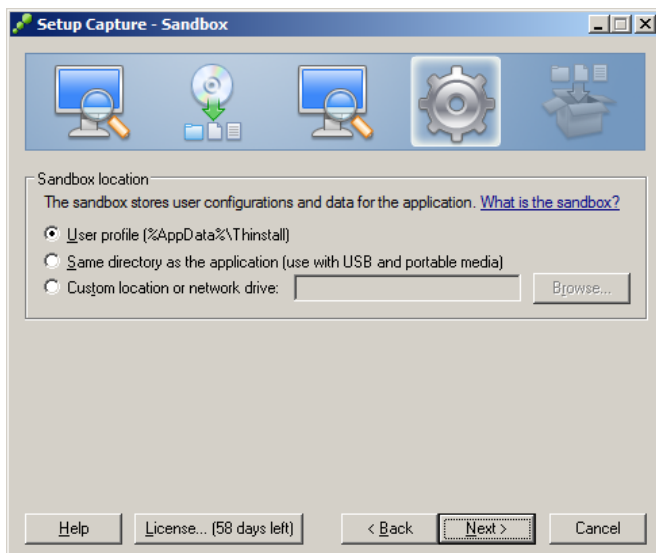
Kuva 7. Sovelluksen käyttöoikeuksien asettaminen ThinAppilla paketoitaessa.

Käyttöoikeuksien jälkeen määritetään, millaiset kirjoitusoikeudet sovelluksella on tietokoneeseen, jossa sitä ajetaan (kuva 8). Sovellus voi saada täydet kirjoitusoikeudet tai rajoitetut kirjoitusoikeudet. Täydet kirjoitusoikeudet antavat sovelluksen kirjoittaa mihin tahansa kansioon asiakastietokoneessa. Rajoitetut kirjoitusoikeudet sen sijaan estävät kirjoituksen suurimpaan osaan kansioista. Muutokset estettyihin kansioihin tallennetaan itse sovellusvirtualisointikuplan sisään. Rajoitettua kirjoitusoikeutta voidaan hyödyntää estämään joitakin hankalasti käyttäytyviä ohjelmia kirjoittamasta kansioihin, joissa muutoksista voisi olla haittaa.



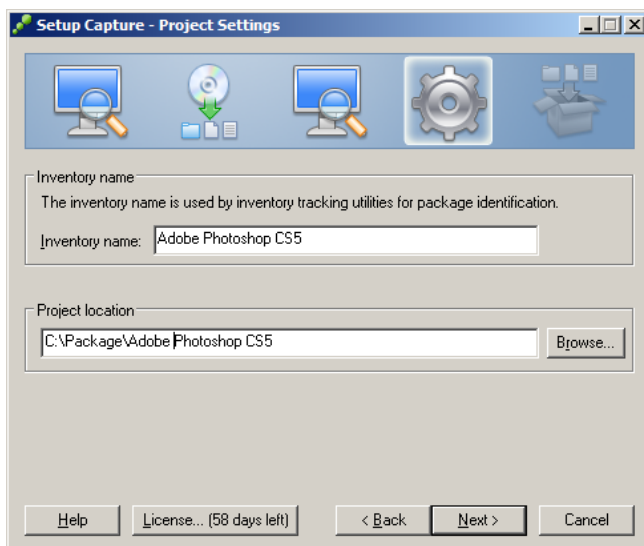
Kuva 8. Sovelluksen kirjoitusoikeuksien valitseminen.

Tämän jälkeen ThinApp-ohjelmistolle määritetään, mihin asiakastietokoneen kansioon sovelluksen asetus- ja datatiedostot tullaan tallentamaan käytön aikana (kuva 9). Kyseiset tiedostot voidaan tallentaa käyttäjän omaan AppData-kansioon, jolloin asetus- ja datatiedostot ovat käyttäjäkohtaisia. Ne voidaan valita tallennettavaksi myös samaan kansioon sovelluksen kanssa, mikä mahdollistaa sovelluksen nopean siirtämisen tietokoneesta toiseen säilyttäen oikeat asetukset. Vaihtoehtoisesti ne voidaan myös tallentaa johonkin itse määritettyyn paikkaan, esimerkiksi verkkolevylle, jolloin ne ovat kaikille käyttäjille ja asiakastietokoneille yhteiset.



Kuva 9. Sovelluksen asetus- ja datatiedostojen tallennuspaikan valitseminen.

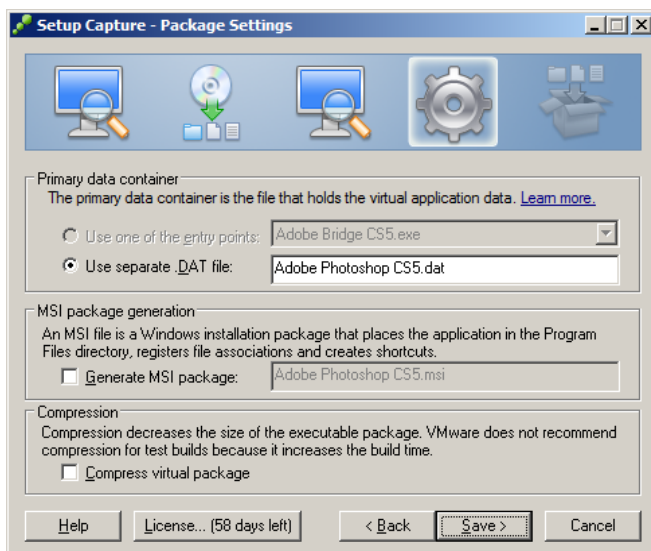
Seuraavassa vaiheessa määritetään, millä nimellä sovellusvirtualisoitua pakettia kutsutaan ja minne se tallennetaan paketoinnin tuloksena (kuva 10).



Kuva 10. Sovellusvirtualisoidun paketin nimeäminen ja tallennuspaikan valitseminen.

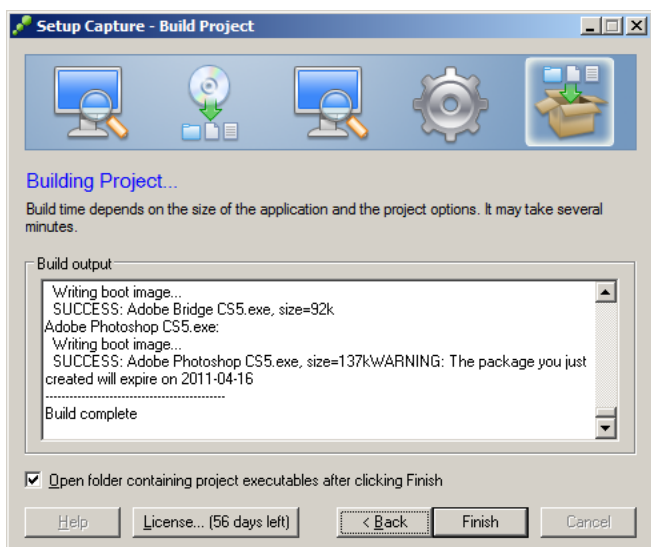
Lopuksi ThinApp-ohjelmistolle määritetään, mihin sovelluksen datatiedostot tallennetaan (kuva 11). Mikäli virtualisoidusta paketista on mahdollista käynnistää useita exe-ohjelmatiedostoja, datatiedostot tallennetaan automaattisesti erilliseen datatiedostoon. Muussa tapauksessa datatiedostot voidaan halutessaan tallentaa myös lopullisen exe-tiedoston sisään. Käyttäjä voi myös valita, luodaanko sovellusvirtualisoidulle sovellukselle asennuspaketti, joka huolehtii tiedostoassosiaatioiden ja pika-

kuvakkeiden luonnista. Sovellusvirtualisoitu paketti voidaan myös pakata, jolloin se vie vähemmän tilaa. Näiden asetusten määrittämisen jälkeen ThinApp kopioi sovellusvirtualisoitavan sovelluksen tiedostot valittuun kohdekansioon ja tekee niistä dat- ja exe-tiedostot, jotka muodostavat lopullisen virtualisoidun paketin.



Kuva 11. Sovelluksen datatiedostojen tallentaminen.

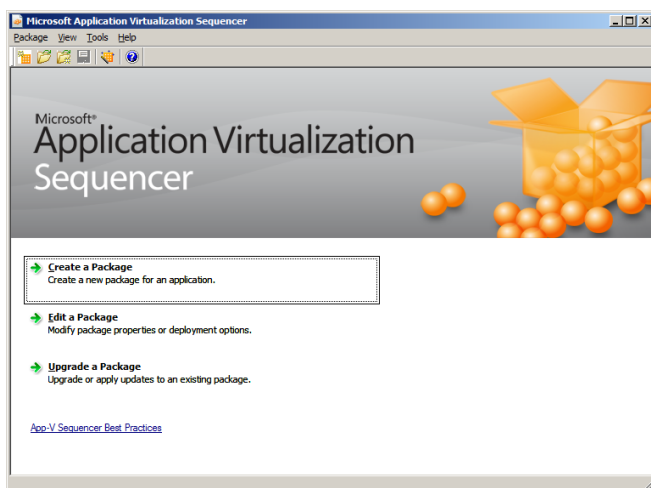
Paketoinnin lopuksi ThinApp ilmoittaa, onnistuiko paketin luonti (kuva 12). Tämän jälkeen sovellus on valmiina käytettäväksi. Sovellus voidaan siirtää toiseen tietokoneeseen kopioimalla vain dat- ja exe-tiedostot, jotka sisältävät kaiken, mitä sovellus vaatii toimiakseen.



Kuva 12. Paketin luomisen onnistuminen

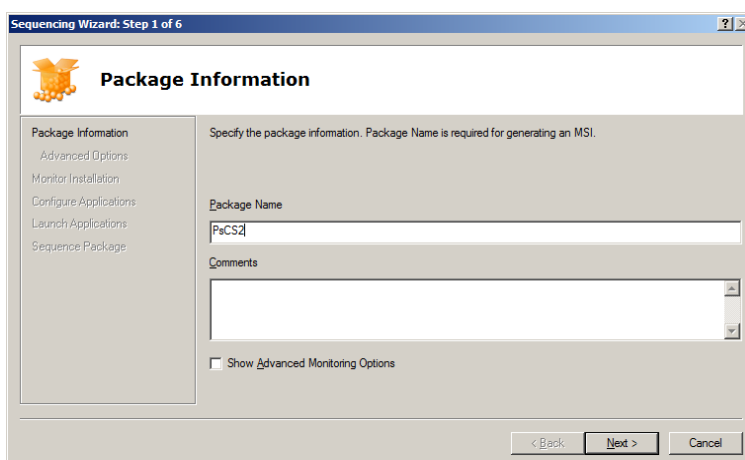
5.2 Paketointi Microsoft App-V -ohjelmistolla

Microsoft App-V:llä paketointia varten tietokoneelle on asennettava Microsoft Application Virtualization Sequencer 4.6 -ohjelma. Ohjelman asennus on nopeaa ja sen käynnistyttyä se on suoraan valmiina paketointia varten. Ohjelman aloitusruutu on esitetty kuvassa 13. Uuden sovelluksen paketointia varten aloitusruudusta valitaan ”Create a Package”.



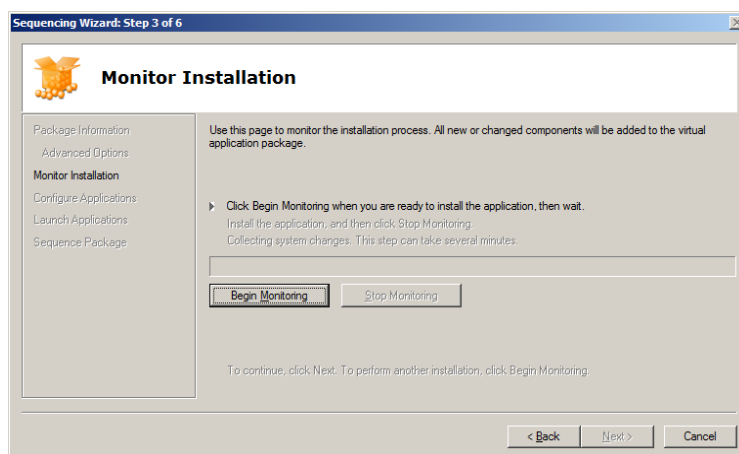
Kuva 13. Microsoft Application Virtualization Sequencer 4.6 -ohjelman aloitusruutu.

Aluksi App-V Sequencer -ohjelmalle määritetään luotavan paketin nimi ja siihen liittyvä mahdollinen kommentti, joka kuvaa sisältöä tarkemmin (kuva 14). On hyvän tavan mukaista, että paketin nimessä käytetään lyhyttä 8.3-tiedostonimeä eli paketin sallittu nimi ennen pistettä saa sisältää maksimissaan 8 merkkiä ja mahdollisen pisteen jälkeen saa olla vain 3 merkkiä (van Bragt 2010).



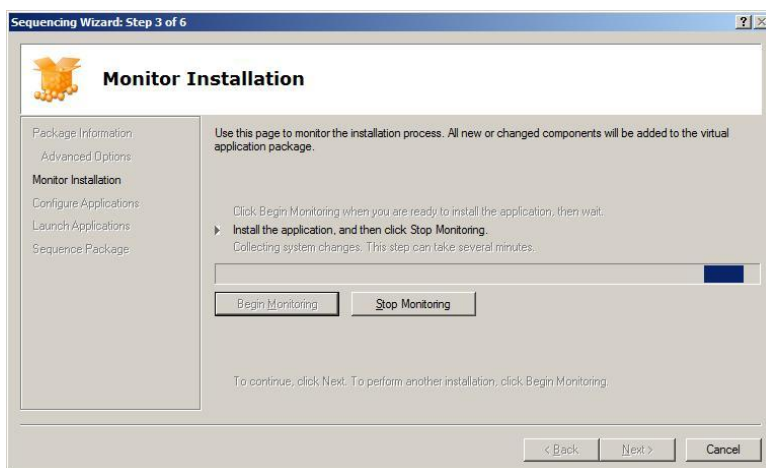
Kuva 14. Paketin perustietojen syöttäminen.

Seuraava vaihe Microsoft App-V Sequencer -ohjelmalla paketoitaessa on seurannan (engl. monitoring) aloittaminen, joka on esitetty kuvassa 15. Tämä vaihe vastaa käytännössä VMware ThinApp -ohjelmiston esiskannausvaihetta. Erona Microsoftin ratkaisussa on kuitenkin se, että seuranta aloitettaessa on määritettävä, mihin kansioon paketoitava sovellus tullaan asentamaan seurannan aikana. Itse sovellusta sinänsä ei ole pakko asentaa kyseiseen kansioon, mutta App-V Sequencer pystyy optimoimaan asennusprosessin seuranta niiden tiedostojen osalta, jotka asennetaan kyseiseen kansioon.



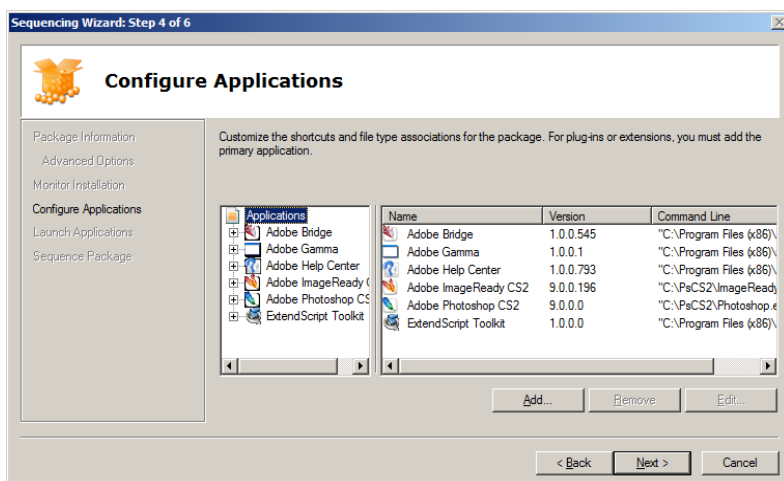
Kuva 15. Seurannan aloittaminen.

Seurannan aloittamisen jälkeen App-V Sequencer on valmis paketoitavan sovelluksen asentamista varten (kuva 16). Kuten ThinApp-ohjelmistonkin tapauksessa, paketoitava sovellus asennetaan aivan perinteisellä tavalla tietokoneeseen. Huomattavaa kuitenkin on, että asennus tulisi mahdollisuuksien mukaan tehdä edellisessä vaiheessa ilmoitettuun kansioon. Sovelluksen asentamisen jälkeen seuranta lopetetaan, jolloin App-V Sequencer kerää kaikki järjestelmässä tapahtuneet muutokset. Tämä vastaa ThinApp-ohjelmiston jälkiskannausvaihetta.



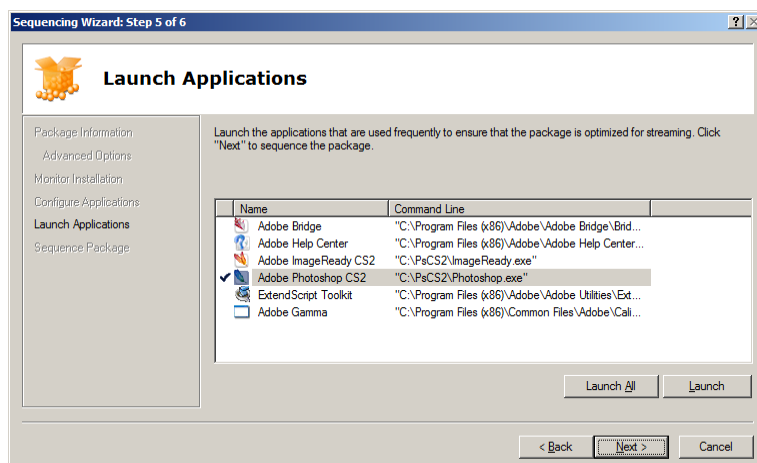
Kuva 16. Microsoft App-V Sequencer valmiina sovelluksen asentamista varten.

Seurannan lopettamisen jälkeen App-V Sequencer -ohjelmalle määritetään, mihin asennettuihin ohjelmiin halutaan luoda pikakuvakkeet ja mitä tiedostoassosiaatioita tulisi rekisteröidä kyseiselle sovellukselle. Tämä vaihe on esitetty kuvassa 17. Pikakuvakkeiden määrittäminen vastaa lähes identtisesti ThinApp-ohjelmiston jälkiskannausvaiheen jälkeen tulevaa vaihetta, jossa valitaan paketin kautta käynnistettävissä olevat exe-tiedostot. Microsoft App-V:n tapauksessa pikakuvakkeet määrittävät, mitkä ohjelmat voidaan paketin kautta käynnistää. Microsoft App-V on kuitenkin ThinApp-ohjelmistoa kehittyneempi siltä osin, että se mahdollistaa myös tiedostoassosiaatioiden määrittämisen, jolloin tiedostoassosiaatiot saadaan suoraan rekisteröityä tietokoneeseen, jossa paketoitua versiota tullaan käyttämään.



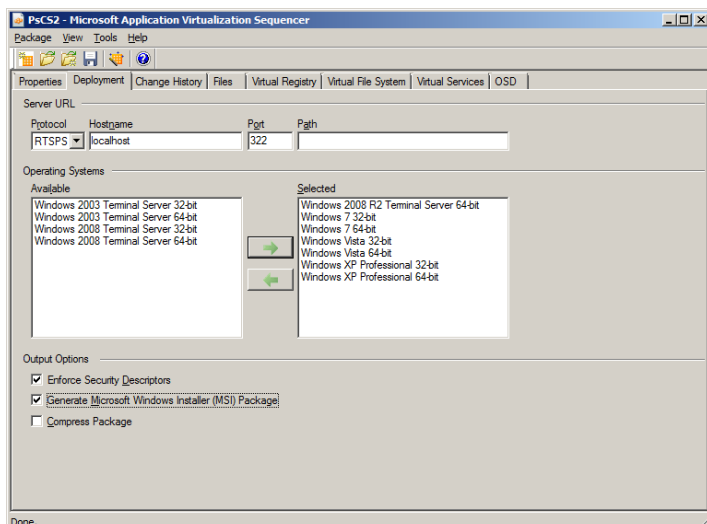
Kuva 17. Pikakuvakkeiden ja tiedostoassosiaatioiden määrittäminen.

Seuraavassa vaiheessa paketin eniten tarvittavat sovellukset voidaan käynnistää paketin optimoimista varten (kuva 18). Sovelluksen käynnistuksen yhteydessä App-V Sequencer tutkii, mitä eri tiedostoja kyseinen sovellus tarvitsee käynnistykseen nopeasti. Näin App-V Sequencer pystyy järjestämään luotavan paketin siten, että sovellusten ensisijaisesti tarvitsemat tiedostot tallennetaan paketin alkuun, jolloin sen siirtäminen suoratoiston avulla verkon yli mahdollistaa sovelluksen käynnistämisen nopeasti jo heti kesken latauksen, kun tärkeimmät tiedostot siirtyvät heti paketin alussa.



Kuva 18. Sovelluksen käynnistäminen paketin optimoimista varten.

Tämän jälkeen App-V Sequencer sarjallistaa sovelluksen valmiiksi paketiksi, jonka jälkeen paketin asetuksia voidaan hienosäätää kuvan 19 mukaisesta näkymästä. Deployment-välilehdeeltä valitaan ne käyttöjärjestelmät, joissa paketin halutaan toimivan. Lisäksi paketille voidaan luoda valmis asennusohjelma laittamalla ruksi kohtaan ”Generate Microsoft Windows Installer (MSI) Package”. Tämä on hyödyllistä etenkin, jos pakettia halutaan testata tai käyttää paikallisesti ilman suoratoistopalvelimen käyttöönottoa. Lopuksi paketti tallennetaan perinteisellä tavalla johonkin haluttuun kansioon.



Kuva 19. Microsoft App-V:llä paketoitun sovelluksen käyttöönottoasetusten määrittäminen.

Microsoft App-V:llä paketoitun sovelluksen käyttöönotto uudessa tietokoneessa vaatii sen, että Microsoft Application Virtualization Client 4.6 -agenttiohjelma asennetaan ensin tietokoneelle. Mikäli paketoituja sovelluksia halutaan asentaa paikallisesti, App-V Client -ohjelma on asennettava toimimaan standalone-tilassa. Tämän jälkeen paketoitu sovellus voidaan asentaa paketoituvaiheessa luodun MSI-asennusohjelman avulla tietokoneeseen. Asennuksen päätyttyä Käynnistä-valikkoon ilmestyy vastaavat pikakuvakkeet, jotka paketoituvaiheessa määritettiin kyseiseen sovellusvirtualisoi-tuun pakettiin. Pikakuvakkeiden kautta käynnistettynä sovellukset näyttävät päällisin puolin käynnistyvän kuten tavalliset, perinteisellä tavalla asennetut sovellukset.

5.3 Käytännön testit ja vertailu

Sovellusvirtualisointiohjelmistojen toimivuuden käytännön vertailua varten valittiin useampi sovellus paketoitavaksi sekä ThinApp- että Microsoft App-V -ohjelmistoilla. Testitapauksina käytettiin Adobe Photoshop -kuvankäsittelyohjelmaa ja OpenOffice.org-toimisto-ohjelmistoa. Kummastakin ohjelmistosta valittiin kaksi eri versiota testattavaksi: Adobe Photoshopin versiot CS2 ja CS5 ja OpenOffice.org-ohjelmiston versiot 2.2 ja 3.3. Adobe Photoshop -testien tarkoituksena oli selvittää, kuinka hyvin sovellusvirtualisointiohjelmistot selviävät hyvin monimutkaisen ja ras-kaan sovelluksen paketoinnista. OpenOffice-testien tarkoituksena sen sijaan oli testa-

ta, pystyykö kaksi eri versiota samasta ohjelmasta toimimaan paketoituna samalla tietokoneella.

Testit suoritettiin siten, että paketointia varten virtuaalikoneeseen tehtiin puhdas asennus Microsoft Windows Server 2008 -käyttöjärjestelmästä, jotta paketointiympäristössä ei olisi mitään ylimääräistä asennettuna. Näin varmistettiin se, että sovellusvirtualisointipaketeista tuli varmasti toimivia. Virtuaalikoneesta luotiin käyttöjärjestelmän asennuksen jälkeen palautuspiste, jotta jokainen paketointi voitiin suorittaa identtisellä käyttöjärjestelmäkonfiguraatiolla. Ennen uuden sovelluksen paketointia virtuaalikone palautettiin siis takaisin puhtaan asennuksen jäljiltä otettuun palautuspisteeseen.

Paketointi suoritettiin molemmilla sovellusvirtualisointiohjelmistoilla edellä kuvatuilla tavoilla. Huomaamisen arvoista on, että kummallakin sovellusvirtualisointiohjelmistolla jätettiin paketin kompressointi tekemättä pakettien kokojen vertailukelpoisuuden takaamiseksi sekä lyhentämään paketointiin kuluva aikaa.

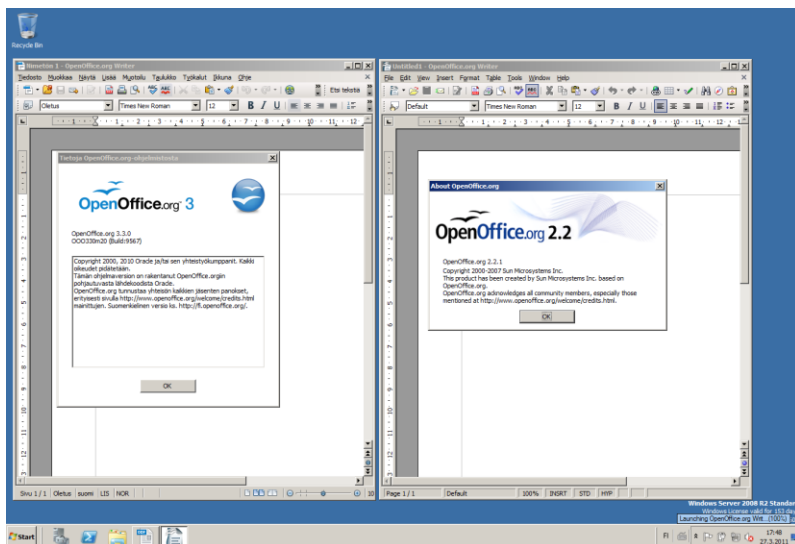
Paketoinnin jälkeen sovellukset testattiin ensin Microsoft Windows Server 2008 -käyttöjärjestelmässä käyttäen puhtaan asennuksen palautuspistettä. Testien kattavuuden ja luotettavuuden takaamiseksi paketoituvat sovellukset testattiin myös Microsoft Windows 7 -käyttöjärjestelmässä, jotta nähtiin, toimiiko sovellusvirtualisointipaketit myös muussa käyttöjärjestelmässä kuin missä ne paketoitiin.

Adobe Photoshop CS5:n paketointi onnistui VMware ThinApp -ohjelmistolla hyvin vaivattomasti. Sovelluksen paketoitu versio toimi hyvin sekä Windows Server 2008:ssa että Windows 7:ssä. Paketointi Microsoft App-V -ohjelmistolla sen sijaan ei onnistunut lainkaan. Adobe Photoshop CS5:n asennusohjelma ilmoitti asennuksen lopuksi, että se ei pystynyt asentamaan kaikkia ohjelmiston vaatimia komponentteja tietokoneeseen. Asennusohjelman pystyi kuitenkin suorittamaan loppuun, mutta paketoitu versio ei suostunut käynnistymään ollenkaan kummassakaan käyttöjärjestelmässä. CS5:n osalta VMware ThinApp toimi siis huomattavasti paremmin kuin Microsoft App-V. Adobe Photoshop CS2:n tapauksessa tilanne oli juuri päinvastainen. VMware ThinApp suoritti paketoinnin onnistuneesti, mutta paketoitun version käynnistämisen yhteydessä Photoshop CS2 ilmoitti, että sen konfiguraatio on vialli-

nen ja se tulisi asentaa uudelleen. Sen sijaan Microsoft App-V:llä paketoitu versio osoittautui toimivaksi.

OpenOffice.orgin versioiden 2.2 ja 3.3 paketointi onnistui ilman mitään ongelmia VMware ThinAppilla. Paketoidut versiot toimivat suoraan sekä Windows Server 2008 - että Windows 7 -käyttöjärjestelmissä. OpenOffice.orgin versio 2.2 toimi Microsoft App-V:llä paketoituna suoraan molemmissa käyttöjärjestelmissä. Sen sijaan OpenOffice.orgin version 3.3 kanssa oli pieniä ongelmia. Sen paketointi onnistui, mutta sitä käynnistettäessä App-V Client ei ymmärtänyt, että käynnistys onnistui ja sulki sovelluksen noin 10 minuutin päästä käynnistyksestä. Ratkaisuna tähän App-V Sequencer -ohjelmassa täytyi tehdä muutos paketin OSD-konfiguraatioon asentamalla virtuaalikoneessa käytettäväksi alijärjestelmäksi ”console” oletusvalinnan ”windows” sijaan. Tämän pienen muutoksen avulla myös OpenOffice.org 3.3 alkoi toimia moitteetta molemmissa käyttöjärjestelmissä.

Molempia OpenOffice.orgin versioita pystyi myös käyttämään yhtäaikaa samassa käyttöjärjestelmässä riippumatta siitä, kummalla sovellusvirtualisointiohjelmistolla OpenOffice.org paketoitiin. Kuvassa 20 on esitetty tilanne, kun molemmat OpenOffice.orgin versiot käynnistettiin yhtäaikaa App-V:llä paketoituna.



Kuva 20. Molemmat OpenOfficen versiot käynnistetty samaan aikaan Microsoft App-V:llä paketoituna.

Eräs vertailun arvoinen asia on virtualisoitujen sovelluspakettien koko. Pienempi koko on aina parempi etenkin, jos sovelluksia on tarkoitus siirtää verkon yli suoratoiston avulla. Virtualisoitujen sovellusten pakettien tiedostokoot eri sovellusvirtualisointiohjelmistoilla virtualisoituna on listattu taulukossa 1.

Taulukko 1. Virtualisoitujen sovellusten pakettien koot.

Sovellukset	Virtualisoidun sovelluksen paketin koko (MB)	
	Microsoft App-V	VMware ThinApp
OpenOffice 2.2	406	463
OpenOffice 3.3	520	595
Adobe Photoshop CS2	563	-
Adobe Photoshop CS5	-	2717

Taulukon 1 mukaan OpenOffice.org-ohjelmistot veivät Microsoft App-V:llä paketoituna 12 % vähemmän levytilaa verrattuna VMware ThinAppin vastaaviin paketteihin. Adobe Photoshopin osalta vertailua ei voitu tehdä luotettavasti, koska kumpakaan Photoshopin versiota ei saatu paketoitua onnistuneesti molemmilla sovellusvirtualisointiohjelmistoilla. Oletettavaa kuitenkin on, että Microsoft App-V tuottaisi pienemmän pakettikoon myös Photoshopin tapauksessa.

Testeistä voi vetää sen johtopäätöksen, että molemmat sovellusvirtualisointiohjelmitot kykenevät paketoimaan perussovelluksia ilman ongelmia. Sen sijaan hyvin monimutkaisten sovellusten, kuten testeissä käytetyn Adobe Photoshopin, paketoinnin onnistuminen riippuu paljolti käytetystä sovellusvirtualisointiohjelmistosta. Mikäli paketointi onnistuu VMware ThinAppilla, ei se välttämättä tarkoita sitä, että se onnistuisi myös Microsoft App-V:llä ja toisinpäin.

Testit osoittivat käytännössä myös sen, että sovellusvirtualisointi mahdollistaa helposti saman sovelluksen eri versioiden asentamisen ja käytön yhtäaikaan samassa tietokoneessa. Sovellusvirtualisointi tuo uusia mahdollisuuksia tietokoneiden ylläpitämiseksi yhteensopivana mahdollisimman monessa tilanteessa.

Testien perusteella VMware ThinApp on myös huomattavasti helppokäyttöisempi kuin Microsoft App-V. Paketointi ThinAppilla onnistuu yleensä ilman manuaalisia muutoksia paketin asetuksiin. Sen sijaan Microsoft App-V voi vaatia hienosäätöä

sovellusten toimivuuden takaamiseksi. Lisäksi suoratoistopalvelimen asennus, jota tässä työssä ei käsitelty tarkemmin, tuo lisää mahdollisia tilanteita, joissa sovellusvirtualisoinnin kanssa voi tulla ongelmia.

5.4 Sovellusvirtualisointiohjelmistojen käytettävyys

Eräs hyvin tärkeä osa ohjelmistojen käyttökelpoisuutta on käytettävyys. Hyvä käytettävyys parantaa työn tuottavuutta sekä vähentää käyttökustannuksia, kun ohjelmisto täyttää käyttäjien tarpeet ja turhan työn osuus vähenee. Lisäksi se pienentää ylläpitokustannuksia takaamalla helppokäyttöisyyden, jolloin vaaditaan vähemmän käyttäjäkoulutusta, käyttötukea ja ylläpitotöitä. (Bias & Mayhew 2005, 17–18.)

Ohjelmistojen käytettävyyttä voidaan arvioida erilaisten heurististen arviointimenetelmien avulla. Eräs näistä on Nielsenin heuristisen arvioinnin lista. Lista koostuu kymmenestä yleisestä periaatteesta, joihin perustuen ohjelmiston käyttöliittymää ja käytettävyyttä voidaan arvioida systemaattisesti. Listan ensimmäinen kohta käsittelee järjestelmän tilan näkyvyyttä. Järjestelmän tulisi aina antaa käyttäjälle palautetta järjestelmän tilasta ja tapahtumista. (Nielsen 2005.) Sekä VMware ThinApp että Microsoft App-V informoivat selkeästi käyttäjää, missä tilassa ohjelmisto on milläkin hetkellä. Kumpikin ohjelmisto kertoo toiminnan etenemisestä sekä ilmaisee selkeästi, mitä missäkin vaiheessa tehdään. Erityisen tärkeää tämä on siinä, että käyttäjä tietää, missä vaiheessa paketoitava sovellus tulee asentaa, ilman, että täytyy tukeutua käyttöohjeeseen. Microsoft App-V antaa järjestelmän tilasta hieman paremmin palautetta kuin VMware ThinApp käyttämällä hyvin paljon edistymispalkkeja, joista pystyy seuraamaan, miten pitkällä jokin vaihe on menossa ja kuinka paljon sen suorittamisesta on vielä jäljellä. ThinAppissa edistymispalkkeja on käytetty huomattavasti vähemmän, eikä esimerkiksi esiskannaus- ja jälkiskannausvaiheissa anneta mitään palautetta siitä, kuinka kauan kyseiset operaatiot kestävät. Lopullista sovellusvirtualisointia pakettia luodessa ThinApp antaa prosentuaalisen arvon paketin luomisen edistymisestä.

Nielsenin listan toinen kohta käsittelee järjestelmän ja todellisen maailman yhteyttä. Käytännössä tämä tarkoittaa sitä, että ohjelmiston käyttöliittymässä tulisi käyttää

käyttäjän luonnollista kieltä ohjelmistoon liittyvien erikoistermien sijaan ja esittää asiat tutussa, loogisessa järjestyksessä. (Nielsen 2005.) Sovellusvirtualisointiin perehtymättömälle ihmiselle VMware ThinAppin ja Microsoft App-V:n kieli ei välttämättä ole kovin helposti ymmärrettävää, koska termistö on hyvin teknistä ja liittyy tiiviisti sovellusvirtualisoinnin peruseräpäätteisiin. Kummassa ohjelmistossa tämä on otettu kuitenkin huomioon. Erityisesti ThinApp selittää kussakin paketoituvaiheessa eteen tulevat termit antamalla yksityiskohtaisemman ja käytännönläheisemmän selityksen, jolloin termit ovat kokemattomallekin käyttäjälle heti ymmärrettävissä. Myös Microsoft App-V:ssä on tiettyjä termejä ja vaiheita selitetty tarkemmin, mutta ei läheskään niin kokonaisvaltaisesti kuin VMware ThinAppissa. Molemmissa ohjelmistoissa asiat esitetään samanlaisessa, hyvin loogisessa järjestyksessä, joka perustuu suoraan sovellusvirtualisoinnin teoriaan.

Kolmas kohta Nielsenin listalla on käyttäjän kontrolli ja vapaus. Ohjelmiston tulee tarjota selkeät poistumistiet ja tukea kumo- ja tee uudelleen -toimintoja. (Nielsen 2005.) Sekä VMware ThinApp että Microsoft App-V antavat käyttäjälle mahdollisuuden keskeyttää paketoituprosessin, jopa kesken esi- tai jälkiskannausvaiheen. Lisäksi molemmat sovellukset tukevat liikkumista dialogin ikkunoissa eteenpäin ja taaksepäin Next- ja Back-painikkeiden avulla. Tältä osin molempien ohjelmistojen käytettävyyden on siis hyvällä tasolla.

Nielsen esittää listansa neljäntenä kohtana johdonmukaisuuden. Tällä tarkoitetaan olemassa olevien standardien ja yhtenäisten merkintätapojen käyttöä läpi koko sovelluksen. (Nielsen 2005.) Kumpikin ohjelmisto käyttää hyvin samankaltaista dialogimaista lähestymistapaa paketointiin. Merkintätavat ovat ThinAppissa hyvin yhtenäiset koko paketoituprosessin ajan. Sen sijaan Microsoft App-V:n johdonmukaisuus kärsii hieman siitä, että dialogimaisen paketoinnin jälkeen paketin asetuksia voidaan muokata tarkemmin välilehtiin perustuvasta käyttöliittymästä, joka eroaa täysin alun dialogimaisesta lähestymistavasta. Tässä vaiheessa kokematon käyttäjä ei enää tiedä, mitä miltäkin välilehdeltä tulisi muokata ja missä järjestyksessä.

Nielsenin listan viides kohta painottaa virheiden välttämistä. Nielsenin mukaan hyvien virheilmoitusten sijaan ohjelmiston tulisi estää virheiden tapahtuminen jo ennalta. (Nielsen 2005.) ThinApp onnistuu tässä erityisen hyvin, koska se selittää eri vaiheis-

sa eteen tulevia termejä yksityiskohtaisesti. Näin ollen käyttäjä ymmärtää paremmin, mitä hän on tekemässä missäkin vaiheessa, ja tekee siten vähemmän virheitä. Microsoft App-V:n osalta tilanne on hieman huonompi, koska alun dialogimaisen lähestymistavan jälkeen käyttäjä pystyy hyvin helposti välilehdistä sotkemaan paketin asetukset siten, että paketti ei välttämättä toimi enää ollenkaan kuten sen pitäisi. Molemmat ohjelmistot kuitenkin edistävät virheiden välttämistä varmistamalla tietyissä isommissa tilanteissa käyttäjältä, haluaako hän todellakin tehdä kyseisen asian.

Kuudentena asiana Nielsen listaa tunnistettavuuden ennen muistamista. Tällä tarkoitetaan sitä, että käyttöliittymässä on pyrittävä välttämään käyttäjän muistin kuormittamista turhaan. (Nielsen 2005.) ThinAppissa tämä on otettu hyvin huomioon näyttämällä vain muutama toiminnallisuus yhtä dialogiruutua kohti. Lisäksi ThinApp ei vaadi, että käyttäjä muistaisi edellisten dialogiruutujen asioita. Microsoft App-V:n dialogimainen lähestymistapa on toteutettu yhtä hyvin kuin ThinAppissa, mutta lopuksi esitettävä välilehtiin perustuva käyttöliittymä esittää ehkä liiankin paljon asioita samalla sivulla, jolloin App-V:n muistettavuus kärsii.

Käytön joustavuus ja tehokkuus on esitetty Nielsenin listan seitsemäntenä kohtana. Edistyneemmille käyttäjille tulisi tarjota oikoteitä sekä mahdollistaa käyttöliittymän mukauttaminen omiin tarpeisiin sopivaksi. (Nielsen 2005.) ThinAppin osalta mukauttamista ei kuitenkaan pysty tekemään, koska paketointiprosessi on luotu täysin kiinteäksi. Toisaalta tämä voidaan nähdä myös hyvänä asiana, koska ThinAppin paketointiprosessi on jo valmiiksi luotu tehokkaaksi ja tiiviiksi. Microsoft App-V kuitenkin tarjoaa enemmän mahdollisuuksia edistyneemmälle käyttäjälle välilehtiin pohjautuvan käyttöliittymän ansiosta. Edistynyt käyttäjä voi ensin paketoita sovelluksen dialogimaisen lähestymistavan avulla ja sen jälkeen muuttaa nopeasti monimutkaisempia asetuksia App-V:n välilehdeltä.

Kahdeksas kohta Nielsenin listalla painottaa yksinkertaisuutta. Dialogeissa ei tulisi siis näyttää ylimääräistä informaatiota. (Nielsen 2005.) ThinAppissa ylimääräisen informaation määrä on minimoitu erittäin hyvin. Samaa voidaan sanoa myös Microsoft App-V:n dialogimaisesta lähestymistavasta. Välilehtiin perustuva käyttöliittymän osa sen sijaan on sekavampi, koska se tarjoaa välillä turhaakin tietoa samalla sivulla.

Nielsenin listan yhdeksäs kohta on virheiden käsittely. Käytännössä tämä tarkoittaa sitä, että käyttäjiä tulisi auttaa tunnistamaan ja ymmärtämään virheitä sekä toipumaan niistä antamalla mahdollisimman selkeät ja yksityiskohtaiset tiedot virheestä ja sen mahdollisista ratkaisuvaihtoehdoista. (Nielsen 2005.) Tämän osalta molemmissa sovellusvirtualisointiohjelmistoissa on parantamisen varaa. ThinApp kertoi kyllä selkeästi paketoinnin epäonnistumisesta, kun sillä yritettiin paketoida 64-bittistä versiota Adobe Photoshop CS5:ta, mutta ratkaisuvaihtoehtoja tilanteeseen ei annettu ollenkaan. Microsoft App-V:n ilmoitus OpenOffice.org version 3.3 käynnistymisen epäonnistumisesta ei antanut myöskään mitään tietoa siitä, miten ongelman voisi ratkaista.

Nielsenin listan kymmenes ja viimeinen kohta käsittelee opasteita ja dokumentaatiota. Sen mukaan ohjelmiston tulisi tarjota selkeät ja konkreettiset ohjeet siltä varalta, että ohjelmiston käyttö ei onnistuisi ilman ohjeita. (Nielsen 2005.) ThinAppissa tärkeimmät vihjeet on sisällytetty suoraan sovelluksen eri dialogiruutuihin, jolloin dokumentaation tarve vähenee huomattavasti. Muu dokumentaatio on julkaistu vain VMwaren sivuilla, eikä siihen pääse käsiksi suoraan paketointisovelluksen kautta, mikä vaikeuttaa tiedon etsimistä. Sen sijaan Microsoft App-V tarjoaa paikallisen ohjetiedoston, johon pääsee käsiksi suoraan paketointiprosessin aikana. Lisäksi ohjeen selaamista on helpotettu huomattavasti sillä, että se avaa sopivan ohjesivun riippuen siitä, missä vaiheessa paketointi on menossa.

Nielsenin käytettävyyden arvioinnin listan perusteella ThinApp on käytettävyydeltään parempi kuin Microsoft App-V. Se tarjoaa suoraviivaisemman ja muistettavamman tavan paketoida sovellukset.

6 SOVELLUSVIRTUALISOINNIN MAHDOLLISUUDET SATAKUNNAN AMMATTIKORKEAKOULUSSA

Satakunnan ammattikorkeakoulu käyttää tällä hetkellä hyvin perinteisiä menetelmiä tietokoneiden ylläpitoon. Ohjelmistojen asennuksessa käytetään etähallintaohjelmistoja ja tietokoneiden asennuksissa mallikoneiden avulla valmiiksi luotuja levykuvia, joiden avulla samaa peruskonfiguraatiota voidaan monistaa useaan tietokoneeseen. Osa ohjelmistoista asennetaan verkon kautta ja osa paikallisesti, jolloin ylläpidon täytyy huolehtia kahdesta erilaisesta päivitysmenetelmästä. Asennuksista koituu ylimääräistä työtä johtuen siitä, että esimerkiksi toimisto-ohjelmistot aiheuttavat ongelmia, koska ne vaativat toimiakseen tietyn selaimen. Lisäksi osa käytetyistä ohjelmistoista on käyttöjärjestelmäriippuvaisia ja Java runtime ei myöskään asennu aina automatisoidusti oikein. Näin ollen Satakunnan ammattikorkeakoulu voi hyötyä sovellusvirtualisoinnin käytöstä huomattavasti.

Sovellusvirtualisoinnin käytölle löytyy paljon mahdollisuuksia Satakunnan ammattikorkeakoulussa. Tietokoneiden määrä on varsin suuri, kun otetaan huomioon sekä IT-luokat että hallinnon yksityisessä käytössä olevat henkilökohtaiset tietokoneet. Tietokoneiden suuresta määrästä johtuen ohjelmistojen asennus vaatii kohtuuttoman paljon resursseja ylläpidolta. Korkeakoulujen resurssit ovat yleensä huomattavasti vaatimattomammat kuin yksityisissä yrityksissä, joten ylläpitohenkilökunnan lisääminen ei ole yleensä realistinen ratkaisu. Sovellusvirtualisoinnin avulla voidaan tehostaa ylläpitoa ilman lisähenkilöstöä, mikä mahdollistaa kustannustehokkaan ylläpidon toteuttamisen Satakunnan ammattikorkeakoulussa.

Satakunnan ammattikorkeakoulussa ohjelmistojen määrä on erittäin suuri. Lisäksi eri tieteenaloilla tarvitaan monenlaisia ohjelmistoja. Pelkästään tietojenkäsittelyn ja tietotekniikan opintoihin liittyviä ohjelmistoja on käytössä hyvin paljon. Muut tieteenalat lisäävät ohjelmistojen kirjoa entisestään, mikä voi tuoda ongelmia ohjelmistojen hallinnoinnin ja ohjelmistoristiriitojen kannalta. Koska oppilaitoksille yleensä ostetaan uusia tietokoneita säännöllisin väliajoin, eri aikakausien tietokoneiden kokoonpanot voivat erota huomattavasti. Erityisesti tämä on nähtävissä siinä, että osassa tietokoneita on käytössä uudempi käyttöjärjestelmäversio kuin vanhemmissa tietokoneissa. Tämä edellyttää sitä, että hankitut ohjelmistot toimivat ilman yhteensopi-

vuusongelmia käyttöjärjestelmäversioista riippumatta. Samalle koneelle täytyy usein asentaa paljon erilaisia ja mahdollisesti myös samojen ohjelmistojen eri versioita, joiden välille syntyy helposti ristiriitoja.

Satakunnan ammattikorkeakoulun valmiudet sovellusvirtualisoinnin käyttöön ovat jo valmiiksi hyvät. Vaikka tietokoneita onkin useassa rakennuksessa, ne on kytketty jo koulun sisäiseen tietoverkkoon. Näin ollen uusia laitteistohankintoja ei tarvitse tehdä tietoverkon rakentamiseksi. Valmiina käytössä oleva tietoverkko mahdollistaa sovellusvirtualisoitujen ohjelmistojen jakelun keskitetysti esimerkiksi suoratoiston avulla.

Satakunnan ammattikorkeakoulujen ohjelmistojen täytyy olla aina ajan tasalla, jotta opetus valmistaisi opiskelijoita parhaiten työelämässä käytettyjen työkalujen käyttöön. Lisäksi yleiset ohjelmat, kuten selain ja sähköpostiohjelma vaativat säännöllisiä päivityksiä luotettavan tietoturvan takaamiseksi. Siksi on hyvin tärkeää, että päivitykset voidaan tehdä usein ja ilman liian suurta vaivannäköä. Koska sovellusvirtualisointi mahdollistaa tehokkaan päivitysten jakamisen siinä missä itse ohjelmienkin asennuksen, sen käyttö Satakunnan ammattikorkeakoulussa tehostaa päivitysten käyttöönottoa oleellisesti.

Satakunnan ammattikorkeakoulussa voi ajoittain tulla tarve muuttaa tietokoneiden ohjelmistokokoonpanoja. Perinteisesti ohjelmistojen lisenssiavaimista täytyy pitää kirjaa, mitkä on jo käytetty ja mitkä käyttämättä, jotta uusiin tietokoneisiin syötetään uudet lisenssiavaimet asennusvaiheessa. Tämä on kuitenkin työlästä, koska lisenssiavaimien hallinnointi täytyy tehdä käsin. Sovellusvirtualisoinnin ja suoratoiston avulla voidaan lisenssien hallinnointi toteuttaa automatisoidusti, jolloin Satakunnan ammattikorkeakoulun ylläpidon täytyy käytännössä pitää huolta vain lisenssien määrittämisestä lisenssipalvelimelle.

Sovellusvirtualisoinnin käyttö ennaltaehkäisee myös opiskelijoiden tekemien virheiden varalta tietokoneissa. Paketoimaton sovellus on paljon haavoittuvampi, koska sen datatiedostoja on hyvin helppo poistaa tietokoneesta. Lisäksi sen asetustiedostoja voidaan helposti muokata käsin. Kummassakin tapauksessa opiskelija voi saada sovelluksen toimimattomaksi ilman suurempaa vaivannäköä. Lisäksi jos tietyt sovellukset sisältävät tunnetusti ohjelmistoristiriitoja, voi käyttäjä jopa tavanomaisin toi-

min saada sovellukset sellaiseen konfiguraatioon, jossa ristiriita tulee taas esiin. Kun sovellus paketoidaan sovellusvirtualisoinnin avulla, sen kaikki asetus- ja datatiedostot sijaitsevat yhden suuren datatiedoston sisällä, jolloin niihin on vaikeampi päästä käsiksi. Lisäksi suoratoisto minimoi kaikki opiskelijan mahdollisuudet sekoittaa jokin sovellus tarkoituksella tai tahattomasti, koska sovelluksen toimiva versio ladataan aina suoratoistopalvelimelta, jonne opiskelijalla ei ole mitään oikeuksia.

Sovellusvirtualisointi mahdollistaa myös uudenlaisen hallinnoinnin opettajien kannettavien tietokoneiden osalta. Sen sijaan että opettajilla olisi henkilökohtaiset kannettavat tietokoneet, koululla olisi joukko kannettavia, jotka olisi valmisteltu sovellusvirtualisointia varten asentamalla niihin vain välttämättömimmät ohjelmistot, kuten palomuri ja virustorjunta, sekä sovellusvirtualisointiratkaisun mahdollisesti vaativa agenttiohjelma. Kyseiset tietokoneet olisivat kaikkien opettajien käytettävissä. Opettajat voisivat esimerkiksi ennen luentoa ottaa minkä tahansa kannettavan tietokoneen mukaan ja suoratoistaa haluamaansa ohjelmaa koulun lähiverkon yli. Kyseiset tietokoneet olisivat kaikkien opettajien käytettävissä. Suoratoiston ansiosta kaikilla tietokoneilla pystyisi ajamaan mitä tahansa koulun suoratoistopalvelimelle asennettuja sovelluksia, jolloin opettajien ei tarvitsisi enää huolehtia ollenkaan siitä, että kaikki hänen tarvitsemansa sovellukset on asennettu kyseiselle tietokoneelle. Omien tiedostojen osalta tällöin olisi järkevintä toimia siten, että opettajilla olisi omat kansiot verkkolevyllä, jolloin tiedostot olisivat aina saatavilla miltä tahansa tietokoneelta koulun lähiverkon kautta. Samalla tiedostojen varmuuskopiointi voitaisiin automatisoida siten, että opettajan ei tarvitse itse huolehtia tiedostojensa varmuuskopioinnista.

Samaa hallinnointiperiaatetta voidaan periaatteessa hyödyntää myös opiskelijoille väliaikaiseen käyttöön annettaviin kannettaviin tietokoneisiin. Tällöin opiskelijoille annettavia tietokoneita ei tarvitse valmistella mitenkään, koska opiskelijan tarvitsemat sovellukset voidaan käynnistää suoratoiston avulla ilman mitään ylimääräisiä asennustoimenpiteitä. Ainoa rajoittava tekijä on se, että opiskelijan täytyy käynnistää sovellukset ensimmäisen kerran koulun verkossa, jotta niistä ehtii latautua tietokoneelle paikallinen versio. Muuten sovellusten käyttö kotona ilman nettiyhteyttä ei olisi mahdollista.

Satakunnan ammattikorkeakoulu pystyy hyödyntämään sovellusvirtualisoinnin mukanaan tuomaa dynaamisuutta myös mahdollisten koulutustapahtumien järjestämisessä. Koska sovellusvirtualisoidut sovellukset voidaan jaella kaikkiin IT-luokkiin suoratoiston avulla, mikä tahansa IT-luokka voidaan valjastaa hyvinkin erilaiseen koulutustarkoitukseen ilman, että luokan valintaa täytyisi tehdä luokan tietokoneisiin asennettuna olevien ohjelmistojen perusteella. Yllättävän koulutustapahtuman organisointi on siis sovellusvirtualisoinnin ja suoratoiston ansiosta toteutettavissa paljon vaivattomammin, koska enää ei tarvitse miettiä, mitkä IT-luokat soveltuvat kyseiseen käyttötarkoitukseen.

Sovellusvirtualisoinnin aikaansaama dynaamisuus helpottaa yleensäkin Satakunnan ammattikorkeakoulun tilanvarausjärjestelmän käyttöä. Koska kaikki IT-luokat soveltuvat mihin käyttöön tahansa, opettajien ei tarvitse luokkatiloja varatessaan pitää huolta siitä, että tarvittavat ohjelmat varmasti löytyvät tietokoneista. Näin tilanvaraus yksinkertaistuu huomattavasti, ja se vähentää eri kurssien organisointiin kuluva aikaa sekä kurssien päällekkäisyyksistä johtuvia ongelmia.

7 SUUNNITTELUNÄKÖKULMAT SOVELLUSVIRTUALISOINNIN KÄYTTÖÖNOTTOON

Eräs hyvin tärkeä osa käyttöönoton suunnittelua on selvittää, mitkä ohjelmistot voidaan paketoita sovellusvirtualisointikuplan sisään. Ohjelmiston lisenssiehdot voivat estää sen käytön virtualisoidussa ympäristössä, ja tämä koskee myös sovellusvirtualisointia (SafeNet). Lisäksi ohjelmiston epästandardinmukainen käyttäytyminen voi aiheuttaa sen, että se ei toimi oikein sovellusvirtualisoituna. Lisenssiehtojen ja yhteensopivuuden tarkastaminen vaatii ylläpidolta paljon resursseja etenkin Satakunnan ammattikorkeakoulun tapauksessa, koska ohjelmistojen valikoima on laaja. Jo pienelläkin ohjelmistojen määrällä tämä on työlästä, koska lisenssiehtojen tulkitseminen voi viedä erittäin paljon aikaa. Lisäksi ohjelmistojen yhteensopivuus voidaan tarkastaa ainoastaan paketoimalla ohjelmisto sovellusvirtualisoinnin avulla ja testaamalla paketoitun version toimivuus perinpohjaisesti. Ohjelmistot täytyy kuitenkin joka tapauksessa paketoita sovellusvirtualisointia varten, joten niiden paketointi yhteensopivuutta tarkastettaessa ei sinänsä ole ylimääräinen toimenpide. Ylimääräistä aikaa kuluu käytännössä vain itse yhteensopivuustestaukseen ja pakotointiin niiden ohjelmistojen osalta, jotka osoittautuvat epäyhteensopiviksi.

Jos halutaan käyttää suoratoistoa ohjelmistojen jakeluun, on sitä varten otettava käyttöön palvelin, joka huolehtii sovellusvirtualisoitujen ohjelmistojen ja niiden lisenssin hallinnoinnista. Suoratoistopalvelimen on suorituskyvyn takia oltava yleensä erillinen kuin organisaation muut palvelimet. Tämä voi tarkoittaa uutta laitteistohankintaa, mikäli tähän käyttöön soveltuvaa ylimääräistä palvelinta ei löydy jo valmiiksi. Palvelimeen on asennettava käyttöjärjestelmän lisäksi valitun sovellusvirtualisointijärjestelmän palvelinpuolen ohjelmisto sekä ne sovellusvirtualisoidut sovellukset, joita halutaan jakaa suoratoiston avulla.

Suoratoistopalvelimen käyttöönotossa on tärkeä tutkia, kuinka ohjelmistojen lisenssiehdot suhtautuvat kyseiseen jakelutekniikkaan. Esimerkiksi Microsoftin tuotteiden lisenssiehdoissa asia tulkitaan siten, että lisenssi täytyy hankkia jokaista sellaista tietokonetta kohti, johon suoratoistopalvelin pystyy toimittamaan kyseisen ohjelman (AppSense 2009). Näin ollen viittä tietokonetta varten on hankittava viisi lisenssiä, vaikka käyttäjiä olisikin todellisuudessa vähemmän. Joidenkin ohjelmistojen tapauk-

sessä tämä voi siis tarkoittaa sitä, että Satakunnan ammattikorkeakoulun olisi hankittava ylimääräisiä lisenssejä pelkästään suoratoistopalvelimen käytön takia. Tällöin viisaampaa onkin toteuttaa tiukemmilla lisenssiehdoilla olevien ohjelmien jakelu käyttämättä suoratoistopalvelinta.

8 YHTEENVETO

Tässä työssä tutkittiin sovellusvirtualisoinnin hyötyjä ja haittoja sekä sen mahdollisuuksia Satakunnan ammattikorkeakoulussa. Työssä keskityttiin vertailemaan kahta sovellusvirtualisointiohjelmistoa, VMware ThinAppia ja Microsoft App-V:tä, ja erityisesti niiden soveltuvuutta Satakunnan ammattikorkeakoulun käyttötarpeisiin.

Sovellusvirtualisointi tarjoaa hyvät mahdollisuudet Satakunnan ammattikorkeakoulun tietokoneiden ylläpidon tehostamiseksi. Se minimoi ohjelmistojen väliset ristiriidat ja mahdollistaa helpon siirtymisen vanhoista käyttöjärjestelmäversioista uudempiin vaarantamatta koulun käytössä olevien ohjelmistojen yhteensopivuutta. Lisäksi se suoraviivaistaa ohjelmistojen jakelua, kun kaikki ohjelmiston vaatimat komponentit voidaan paketoita saman sovellusvirtualisoidun paketin sisään. Näistä hyödyistä johtuen Satakunnan ammattikorkeakoulun säästöt ylläpitokuluissa voivat olla sovellusvirtualisoinnin käyttöönoton myötä varsin merkittävät.

Ohjelmistojen paketointi sovellusvirtualisoinnin avulla ei kuitenkaan ole aina ongelmattonta. Osaa ohjelmistoista ei pystytä paketoimaan, jolloin ne täytyy asentaa perinteisellä tavalla. Sovellusvirtualisoinnin hyödyt riippuvatkin paljolti siitä, miten suuri osa ohjelmistotarjonnasta on mahdollista paketoita onnistuneesti.

Testien perusteella VMware ThinApp on varteenotettavampi vaihtoehto kuin Microsoft App-V sovellusvirtualisoinnin toteuttamiseksi Satakunnan ammattikorkeakoulun Windows-ympäristössä. ThinApp on helppokäyttöisempi ja sen avulla on nopeampi paketoita ja testata pakettien toimivuus. Lisäksi pakettien luominen ThinAppilla vaatii vähemmän manuaalisia muutoksia pakettien konfiguraatioon.

Jatkotutkimuksissa olisi hyvä selvittää, kuinka hyvin sovellusvirtualisointi saadaan toimimaan suoratoiston avulla lähiverkossa, joka sisältää useita tietokoneita. Tämä voitaisiin toteuttaa laajemman käyttöönottotestin muodossa. Samalla tulisi ilmi mahdollisia käytännön ongelmatilanteita, kun sovellusvirtualisointia otetaan käyttöön isoissa organisaatioissa.

Lisäksi olisi syytä tutkia, kuinka hyvin Satakunnan ammattikorkeakoulun koko käytössä oleva ohjelmistotarjonta voidaan paketoida sovellusvirtualisoinnin avulla. Näin saataisiin selvitettyä, miten suuri osa koulun ohjelmistoista on ylipäättään mahdollista virtualisoida onnistuneesti. Tämän tiedon perusteella voitaisiin tarkemmin tehdä arvio siitä, kuinka hyvin sovellusvirtualisoinnin käyttöönotto onnistuisi Satakunnan ammattikorkeakoulussa.

LÄHTEET

4Sysops. 2008. The disadvantages of Virtualization. Viitattu 8.12.2010. <http://4sysops.com/archives/the-disadvantages-of-application-virtualization/>

AppSense. 2009. Microsoft Application License Control in Virtual Environments. Viitattu 14.2.2011. [http://www.appsense.com/Files/Documents/Microsoft %20Application%20License%20Control%20\(US\).pdf](http://www.appsense.com/Files/Documents/Microsoft%20Application%20License%20Control%20(US).pdf)

Bias, R. G. & Mayhew, D. J. 2005. Cost-Justifying Usability: An Update for the Internet Age. Second Edition. USA: Morgan Kaufmann.

Cameyon www-sivut. Viitattu 16.2.2011. www.cameyo.com

Competitive System Analysis. 2009. Application Virtualization 2008-2009. Assessing the Architectural and Performance Characteristics of Four Leading Windows Application Virtualization Solutions. Viitattu 20.3.2011. <http://communities.vmware.com/servlet/JiveServlet/download/115114118140/xpnet%20Performance%20Review%20of%20AppVirt%20Solutions%5B1%5D.pdf>

Elhajj, T. 2005. Deploying Microsoft Windows Server Update Services. Viitattu 2.4.2011. <http://www.microsoft.com/downloads/en/confirmation.aspx?familyid=e99c9d13-63e0-41ce-a646-eb36f1d3e987&displaylang=en>

Hoppe, M & Seeling, P. 2010. Security of Virtualized Applications: Microsoft App-V and VMware ThinApp. Teoksessa T. Sobh & K. Elleithy (toim.) Innovations in Computing Sciences and Software Engineering. USA: Springer Science+Business Media B.V, 291-295.

Jackson, C. Making applications compatible with Windows 7 in a Virtualized environment. Viitattu 8.12.2010. <http://technet.microsoft.com/enus/magazine/ff458340.aspx>

Mann, A. 2006. Virtualization 101: Technologies, Benefits and Challenges. Viitattu 22.11.2010. [http://www.etomicmail.com/files/dedicated_server/Virtualization %20101.pdf](http://www.etomicmail.com/files/dedicated_server/Virtualization%20101.pdf)

Microsoft. 2011. Microsoft Application Virtualization Technical Overview. Viitattu 31.1.2011. <http://www.microsoft.com/systemcenter/appv/techoverview.mspx>

Muller, J. N. 1998. Desktop encyclopedia of telecommunications. New York: Mc Craw-Hill.

Mäntylä, J.-M. 2008. Virtualisointi mullistaa tietotekniikan. Viitattu 22.9.2010. <http://www.tietoviikko.fi/cio/article192316.ece>

Nielsen, J. 2005. Ten Usability Heuristics. Viitattu 2.5.2011. http://www.useit.com/papers/heuristic/heuristic_list.html

Petrović, T. & Fertalj, K. 2009. Demystifying desktop virtualization. Teoksessa Proceedings of the 9th WSEAS international conference on Applied computer science. USA: WSEAS, 241-246. Viitattu 30.3.2011. <http://www.wseas.us/e-library/conferences/2009/genova/ACS/ACS-40.pdf>

Ruest, D. & Ruest, N. 2009. Virtualization: Beginner's guide. New York, Chicago: McCraw-Hill.

SafeNet. 2010. Application Virtualization and Software Licensing: Best Practices for Software Vendors. Viitattu 5.4.2011. <http://www.softwaremag.com/linkservid/1D3C7446-F70B-E644-9EFBF83A35F212B0/showMeta/0/>

Turkulainen, J. 2004. Remote Library Injection. Viitattu 8.12.2010. <http://www.nolugin.org/Downloads/Papers/remote-library-injection.pdf>

van Bragt, W. 2010. App-V Sequencing. Viitattu 18.3.2011. <http://www.virtualizationadmin.com/articles-tutorials/application-virtualization-articles/app-v-sequencing.html>

VMware. 2007. Profile Virtualization: A Guide to understanding profile management with VMware View. Viitattu 30.3.2011. <http://www.vmware.com/files/pdf/resources/vmware-view-profile-virtualization.pdf>

VMware. 2010a. Viitattu 16.9.2010. <http://www.vmware.com/products/thinapp/overview.html>.

VMware. 2010b. ThinApp User's Guide – ThinApp 4.5. Viitattu 3.4.2011. http://www.vmware.com/pdf/thinapp45_manual.pdf